ÑO II - NUM. 51

MICROHOBBY

REVISTA INDEPENDIENTE PARA USUARIOS DE ORDENADORES SINCLAIR

SEMANAL

AÑO II - N.º 51

125 PTS.

Canarias 135 ptas.

EDITA HOBBY PRESS S.A



HARDWARE

MICROPROCESADOR Z-80 EL CORAZON DEL SPECTRUM

UTILIDADES

UNA RUTINA PARA "FRENAR" LA VELOCIDAD DE LOS PROGRAMAS

PROFESOR PARTICULAR

UNA AYUDA INDISPENSABLE PARA TUS ESTUDIOS

EXCLUSIVA

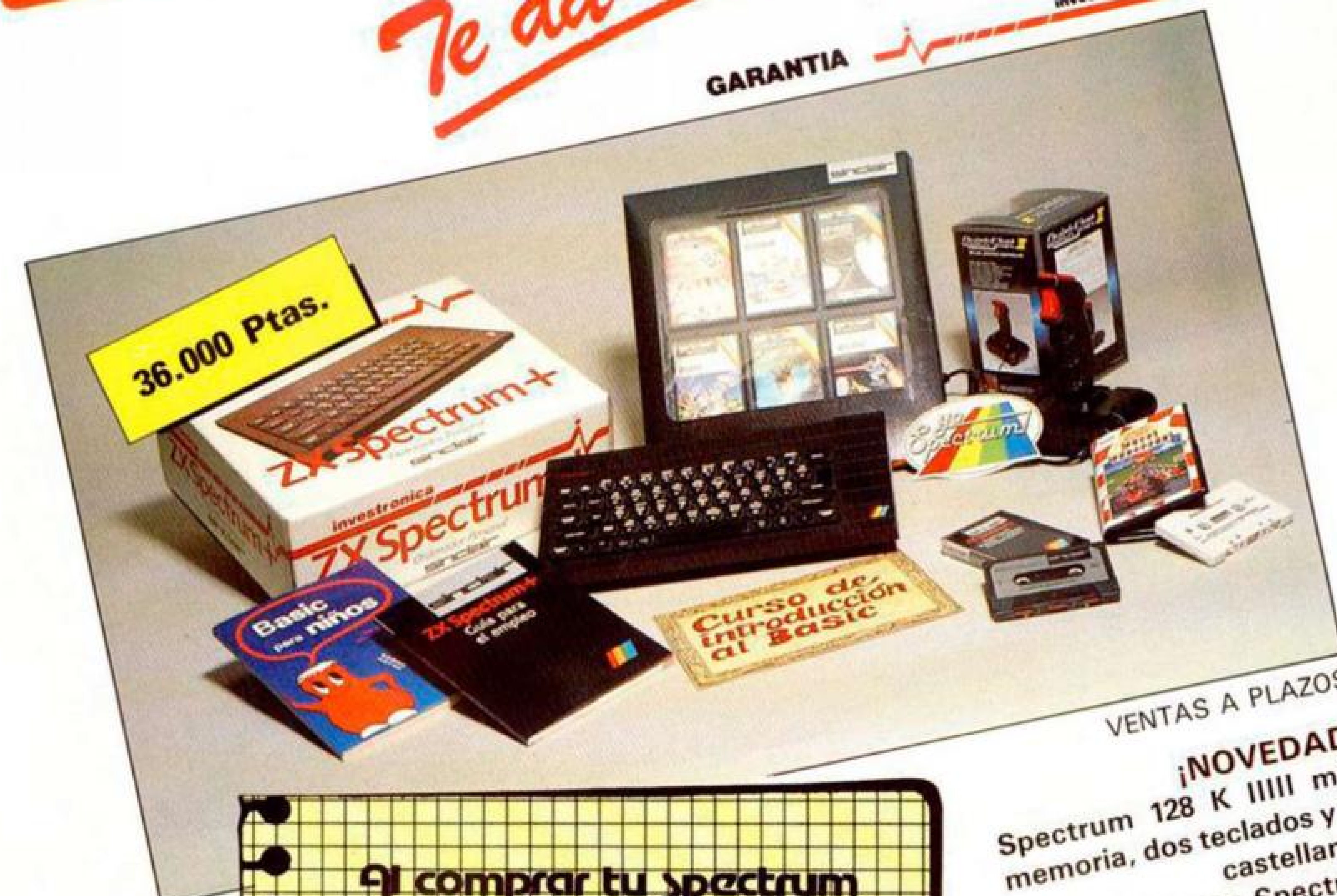
LAS PRIMERAS FOTOS DE "SHADOW OF THE UNICORN"

CONTIENE 4 FICHAS COLECCIONABLES DE CODIGO MAQUINA

COMPUTIQUE

Te da más

GARANTIA investronica



VENTAS A PLAZOS

¡NOVEDAD!

Spectrum 128 K IIIII más memoria, dos teclados y en castellano.

Compatible con Spectrum 48 K/Plus

¡Más de 5.000 programas!

Al comprar tu spectrum

te regalamos

- Joystick Gunshot I
- Estuche con seis juegos originales
  - Backgammon
  - Reversi
  - Planetoids
  - Cookie
  - Psst
  - Jet Pac
- Un estupendo libro de Basic para niños
- Y además te obsequiamos con un curso de introducción al Basic



Embajadores, 90 Tfno. 2270980
28012 Madrid

Servimos a tiendas
Abrimos sábados por la tarde

COMPUTIQUE

Director Editorial
José I. Gómez-Centurión

Director Ejecutivo
Domingo Gómez

Asesor técnico editorial
Gabriel Nieto

Redactor Jefe
Africa Pérez Tolosa

Diseño
Rosa María Capitel

Redacción
Amalio Gómez,
Pedro Pérez
Jesús Alonso

Secretaria Redacción
Carmen Santamaría

Colaboradores
Primitivo de Francisco,
Rafael Prades,
Miguel Sepulveda,
Sergio Martínez y J. M. Lazo

Fotografía
Javier Martínez, Carlos Candel

Portada
José María Ponce

Dibujos
J. R. Ballesteros, A. Perera,
F. L. Frontán, Pejo, J. M. López
Moreno J. Igual, J. A. Calvo, Lóriga,
J. Olivares

Edita
HOBBY PRESS, S. A.

Presidente
María Andrino

Consejero Delegado
José I. Gómez-Centurión

Jefe de Publicidad
Marisa Esteban

Publicidad Barcelona
José Galán Cortés
Tels.: 303 10 22 - 313 71 76

Secretaría de Dirección
Marisa Cogorro

Suscripciones
M.ª Rosa González
M.ª del Mar Calzada

Redacción, Administración y Publicidad
La Granja, s/n
Polígono Industrial de Alcobendas
Tel.: 654 32 11
Telex: 49480 HOPR

Dto. Circulación
Carlos Peropadre

Distribución
Coedis, S. A. Valencia, 245
Barcelona

Imprime
Rotedic, S. A. Ctra. de Irún,
km. 12,450 (MADRID)

Fotocomposición
Espacio y Punto, S. A.
Paseo de la Castellana, 268

Fotomecánica
Grof
Ezequiel Solana, 16

Depósito Legal
M-36.598-1984

Representante para Argentina, Chile, Uruguay y Paraguay, Cía. Americana de Ediciones, S.R.L. Sud América 1.532. Tel.: 21 24 64. 1209 BUENOS AIRES (Argentina).

Solicitado control
OJD

MICROHOBBY ESTA SEMANA

AÑO II. N.º 51. 5 al 11 de noviembre de 1985
125 ptas. (Sobretasa Canarias 10 ptas.)





Fotos en exclusiva de «Shadow of the Unicorn»

PREMIADOS HOBBY-SUERTE

JOSE MOLINA FERNANDEZ. Huertas, 8. Macael (ALMERIA).
Cinta de programas (5.ª Cat.)

RAFAEL DEL OLMO ARANZO. Nuria, 87, 1.º C. (MADRID).
Cinta de programas (5.ª Cat.)

JAVIER MARTINEZ FRIAS. Goya, 19. (CADIZ).
Cinta de programas (5.ª Cat.)

PEDRO ALONSO LLORENTE. Avda. Sancho el Fuerte, 96, Esc. 7.º B. (PAMPLONA).
Cinta de programas (5.ª Cat.)

DAMIAN GARCIA MOLINA. Antonio Mora Fernández, 26, 5.º Elche (ALICANTE).
Cinta de programas (5.ª Cat.)

JORDI CANALS GRAU. Aguda, 38, 5.º A. (GERONA).
Cinta de programas (5.ª Cat.)

PABLO MEDIANO DE ARRIBA. Narváez, 14, 6.º Iz. (MADRID).
Cinta de programas (5.ª Cat.)

FCO. Y CARLOS LOPEZ RIVERO. Pso. de Yeserías, 11, 6.º C. (MADRID).
Impresora GP 50 (2.ª Cat.)

JOSE RAMON TOMAS LLACUNA. Aulestía i Oijoan, 24. Entr. 1.º (BARCELONA).
Cinta de programas (5.ª Cat.)

RICARDO SANCHEZ GARCIA. Corregidor Juan Fco. de Luján, 78 (MADRID).
Cinta de programas (5.ª Cat.)

JOSE RAGES MUÑARACH. Muns, 19. Hospitalet (BARCELONA).
Cinta de programas (5.ª Cat.)

MICROPANORAMA

SHARP 10-0700

UNA IMPRESORA A TODO COLOR

En la era de la automatización SHARP acaba de desarrollar una silenciosa impresora de inyección con tinta de colores. Usando la más alta tecnología, esta impresora puede llegar a reproducir en siete colores, lo que la hace imprescindible para todo tipo de grabados.

Acoplándola a una computadora personal, puede ser de gran utilidad para negocios, educación, topografía computerizada y cualquier hobby. La impresora SHARP 10-0700 tiene un precio de 290.000 pesetas y desarrolla una muy alta resolución.

Los resultados están a la vista.





SHATTER, El primer comic escrito, dibujado e ideado por un ordenador

Mike Saenz, un joven artista americano, es el creador de este innovador comic, cuyo argumento y dibujos han sido realizados con la ayuda de un ordenador.

La trama de este comic y de la futura serie que se está preparando basada en este personaje, se sitúa en pleno siglo XXI, en una ciudad llamada Chicagoland, y narra las peripecias de un joven buscador de recompensas —Shatter—, que descubre una organización clandestina llamada Executariant, dedicada a robar la mente de las personas.

Saenz ha utilizado para su creación un ordenador Macintosh de Apple, además de un digitalizador de imágenes MacVision que le permite transformar las imágenes obtenidas por un vídeo o cámara a pixels y llevarlos a continuación a la pantalla de su ordenador.

¿Sustituirán algún dia los ordenadores a los artistas?

Sobre este tema Saenz piensa que los ordenadores ayudan a dibujar, pero que no lo hacen todo por ti, pues un ordenador no es un artista. Nosotros compartimos esta opinión.



LAS PRIMERAS FOTOS DEL «SHADOW OF THE UNICORN»



¡Por fin lo tenemos en nuestras manos! Nuestra «hada madrina» particular nos lo acaba de enviar y raudos hemos corrido hacia el ordenador dispuestos a disfrutar con esta pequeña maravilla de la programación.

Hemos estado durante más de quince minutos recorriendo caminos y veredas, atravesando bosques y montañas, cruzando ríos, recogiendo objetos, luchando contra los enemigos, descubriendo parajes inóspitos y visitando castillos encantados..., y cuál ha sido nuestra sorpresa cuando tras morir uno de los personajes —os recordamos que se pueden controlar a 10 diferentes—, aparece en la pantalla el mensaje: «Ha realizado el 1% del juego».

—¡Pero bueno! —hemos exclamado al unísono— ¿solamente el 1%?

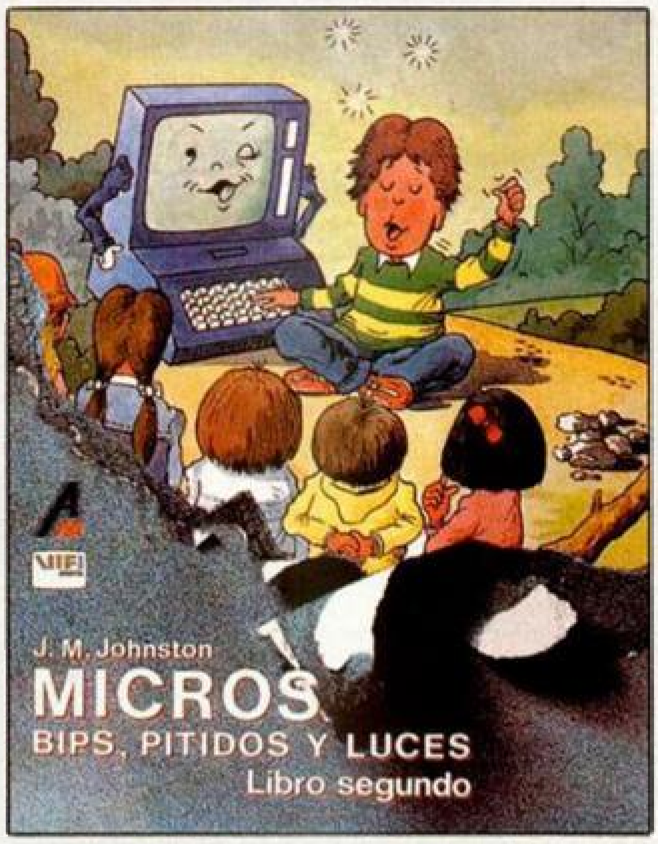
Creo que con esto ya os decimos bastante a cerca de este fabuloso «Shadow of the Unicorn», aunque lamentablemente no podemos comentaros por el momento mucho más, pues ni nosotros mismos tenemos aún muy claro qué es lo que hay que hacer para completar el juego, y ya tenemos a nuestro experimentado equipo de «viciosos del joystick» trabajando intensamente en esta difícil tarea.





No os pongáis nerviosos, pues seguro que pronto, muy pronto, «Shadow of the Unicorn» saldrá a la venta y todos tendremos la oportunidad de adquirir este brillante juego.

LIBROS



MICROS: BIPS, PITIDOS Y LUCES

J. M. Johnston. Anaya Multimedia, 96 págs.

¿Cómo funciona un ordenador?, ¿qué es la CPU?, ¿cómo es la memoria? Ahora los más pequeños —fundamentalmente los comprendidos entre los siete y trece años de edad— tendrán la oportunidad de encontrar la respuesta a todas estas preguntas básicas, gracias a este entretenido libro escrito por J. M. Johnston y editado en España por Anaya Multimedia.

La principal finalidad de este «Micro: Bips, pitidos y luces», es la de acercar a los niños al mundo de la informática a través del conocimiento y el aprendizaje de una serie de conceptos básicos del funcionamiento de los ordenadores: como cuentan, porqué son capaces de realizar las operaciones con tanta rapidez, qué sistema de numeración utilizan..., además de servirles como toma de contacto con temas como los lenguajes de programación, los diferentes tipos de programas o los diagramas de flujo.

El autor de este libro ha conseguido imprimirle una gran agilidad de lectura y una fácil comprensión, a través de un supuesto diálogo entre un niño y la persona que le está enseñando, todo ello acompañado de un buen número de dibujos.

SINCLAIR AUMENTA SU CUOTA DE MERCADO BRITANICO AL SESENTA POR CIENTO

La participación de Sinclair Research en el mercado británico de los ordenadores personales ha crecido en un 20 por 100 en los últimos meses hasta situarse en un 56,8 por 100, según ha anunciado la compañía de investigación de ventas AGB.

Sinclair está vendiendo actualmente el doble de ordenadores que el conjunto de sus inmediatos competidores, Commodore y Acorn, siendo el modelo ZX Spectrum Plus el más adquirido.

Alison Maguire, directora de Marketing de Sinclair, ha señalado que estos excelentes resultados se han producido cuando aún no se ha entrado en la temporada navideña, que es cuando hay un mayor número de ventas.

Según los datos facilitados por AGB, este 56 por 100 del mercado se divide en un 50,5 por 100 para el modelo Spectrum Plus, un 5,8 para el QL y un 0.5 por 100 para el resto de aparatos de la compañía.

VISITA RELAMPAGO DE TRAMIEL A ESPAÑA

En fechas muy recientes, el director de la empresa de ordenadores Atari, Jack Tramiel, ha realizado una rápida visita a nuestro país con motivo de mantener ciertas negociaciones con los directivos de Investrónica, actual distribuidora de Sinclair en España.

Parece que estas conversaciones han resultado satisfactorias para ambas partes y que se ha llegado a un acuerdo mediante el cual Investrónica será, dentro de un breve periodo de tiempo, la encargada de la distribución de los productos Atari en España.

Entre estos productos destaca el modelo 520 ST, la actual estrella de Atari, cuyo precio en el mercado será inferior a las 200.000 pesetas, incluyéndose en el lote un monitor en blanco y negro, ratón y unidad de disco.



Una oferta que vale por... 5

AL REALIZAR TU SUSCRIPCION A MICROHOBBY RECIBIRAS TOTALMENTE GRATIS UNA SELECCION DE FABULOSOS PROGRAMAS ELEGIDOS ENTRE LOS DE MAYOR EXITO DE TU FIRMA DE SOFTWARE FAVORITA: US GOLD

Blue Max
Tienes los controles de un avión de la primera guerra en tus manos. Derriba al enemigo, bombardea blancos, destruye tanques y emplazamientos de armas.

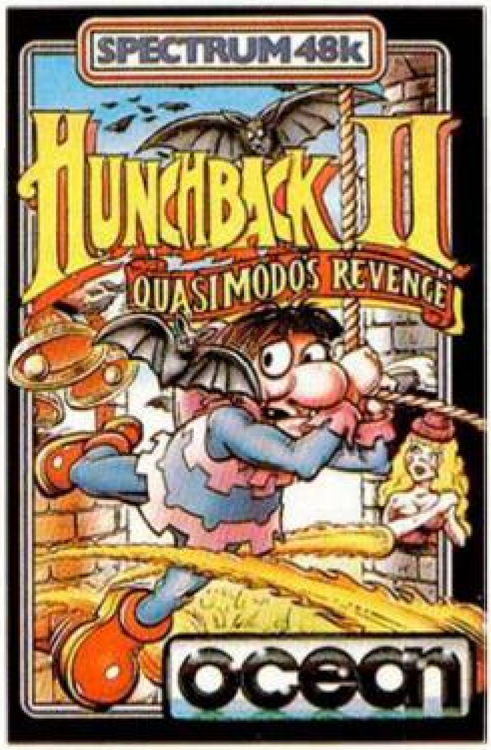


Estos cinco programas, que han encabezado listas de éxitos de toda Europa durante este año, están contenidos en **dos cintas** de cassette para Spectrum. De esta forma van a aparecer en breves días en el mercado inglés, como gran oferta de Navidad de la firma US GOLD, bajo el nombre genérico de **«Arcade Hall of Fame».** En España se pondrán a la venta al precio de 2.500 ptas., aunque el valor total de los cinco programas es muy superior.

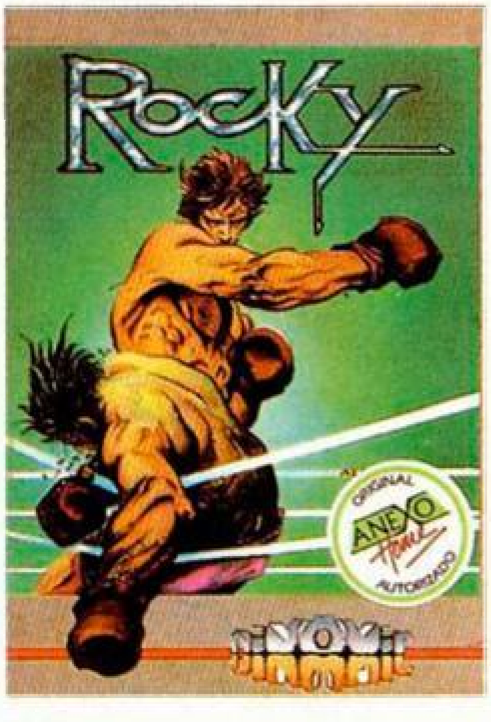
Microhobby Semanal te regala ahora estos cinco estupendos programas, al realizar tu suscripción, sólo hasta el próximo 31 de diciembre.

Lo increíble de esta oferta-regalo es que su valor es casi el 50 por 100 del precio real de suscripción.

Suscribirse a **Microhobby** es el sistema ideal para recibir nuestra Revista puntualmente y para ahorrar mucho dinero en su precio. Si consideras, además, el valor de los programas que recibes de regalo, no nos cabe duda de que preferirás recibir en tu casa **Microhobby** el próximo año.

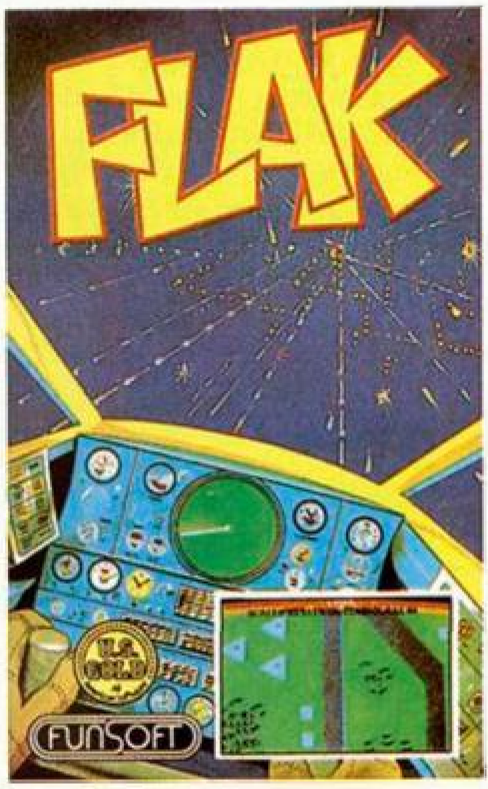


Hunchback II
Quasimodo busca a Esmeralda prisionera en el castillo. Durante las seis primeras imágenes debes accionar la campana gigante, hasta llegar a la liberación de Esmeralda.

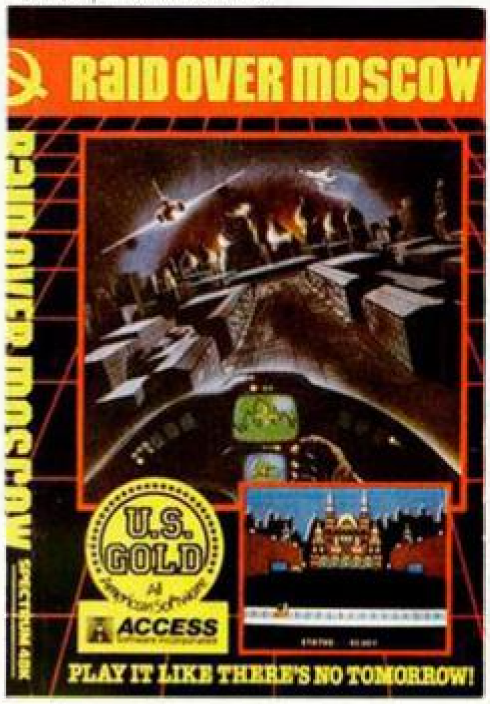


Rocky
Vive la emoción de un encarnizado combate de boxeo, con Rocky. Tendrás que pelear duro para conseguir el campeonato del mundo; sólo los puños de tu oponente se interponen en tu camino hacia el triunfo final.

Flak
En el año 2096 el universo está controlado por ordenadores. Una fuerza siniestra que quiere eliminar la libertad. Tu misión como piloto guerrero es atacar y destruir esta amenaza.



Raid Over Moscow
Defiende a USA y Canadá del ataque nuclear que ha lanzado Rusia contra ellos. Con tu escuadrilla habrás de hacer un viaje lleno de peligros hasta llegar al mismísimo Kremlim y destruir las bases de lanzamiento soviéticas. Gráficos y acción sensacionales.



ENVIA HOY MISMO EL CUPON QUE ESTA COSIDO AL FINAL DE LA REVISTA Y ELIGE LA FORMA DE PAGO QUE TE RESULTE MAS FAVORABLE

SPACESHIP

Julio RIBERA

Spectrum 48 K

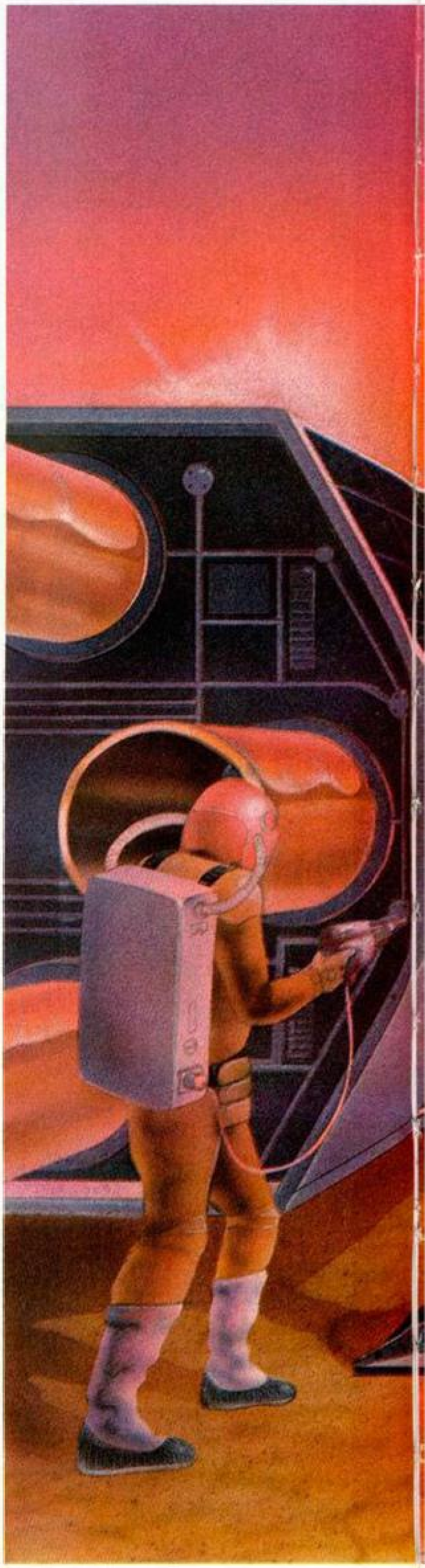
Una sensación de nerviosismo se ha apoderado de nosotros ante la apremiante situación que se nos avecina. Nuestro planeta está a punto de explotar y la única salvación, una nave, tiene problemas.

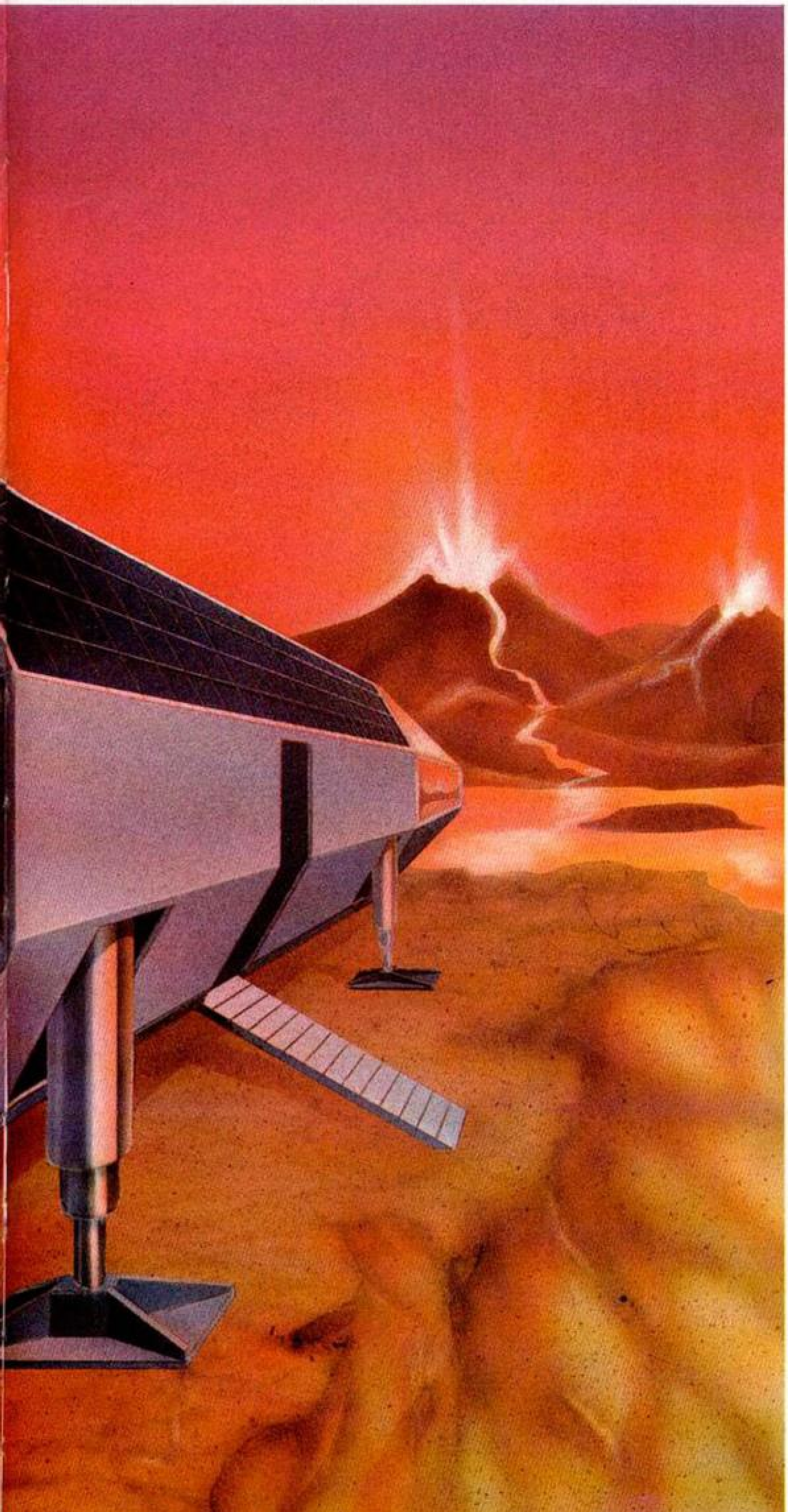
Tendremos que hacernos con las diez tuercas que necesita el aparato para poder despegar que, una vez en nuestras manos, tendremos que colocar hasta conseguir que el marcador de energía se ponga intermitente. Pero en nuestro ansiado deseo de abandonar este infernal planeta, lucharemos también con la infinidad de peligros que nos rodean: arañas acechantes que intentarán deborarnos, lagunas de ácidos que acabarán con nosotros al menor contacto, y, sobre todo, la escasez de tiempo con que contamos para huir antes de la destrucción total.

Como comprobaréis vosotros mismos, la velocidad de movimientos será la única medida posible para llegar a alcanzar un planeta más tranquilo donde sobrevivir.

```
  10 PAPER 0: BORDER 0: CLS : GO
 SUB 6900
  15 CLEAR 51455
  20 DIM A$(16,4,4): DIM B$(8,4,
2): DIM C$(3,5)
  25 PRINT AT 10,10; INK 4;"ESPE
RA UN MOMENTO"
  30 GO SUB 7000
  40 GO SUB 8000
  45 PRINT AT 12,10; INK 2;"PULS
A UNA TECLA"
  50 PAUSE 0: CLS
  55 POKE 23607,200
  60 POKE 23674,0: POKE 23673,0:
 POKE 23672,0
  62 LET NT=0: LET NH=3: LET U=0
  64 LET T=0: LET TH=99999999
  66 LET LN=12: LET Z=0: LET D=1
  67 LET C$(1)=" STU ": LET C$(2
)="  SUU": LET C$(3)="STU  "
  68 RESTORE 75
  70 FOR M=1 TO 8
  72 FOR N=1 TO 4
  73 READ B$(M,N)
  74 NEXT N: NEXT M
  75 DATA "89",":;","<=",">?",CH
R$ 32+CHR$ 33,CHR$ 34+CHR$ 35,"$
%","&'","89",":;","<=",">?","HI"
,"JK","LM","NO","()","*+",",-","
./","01","23","45","67","8A","BC
","DE","FG","01","23","45","67"
  80 LET L=12: LET C=12: LET NP=
1
  81 POKE 23607,60: PRINT INK 4;
AT 19,24;"ENERGY": POKE 23607,20
0
  82 GO SUB 5950
  84 GO SUB 5500
  90 GO SUB 5600
 100 GO SUB 6100
 110 GO SUB 5900
 120 GO SUB 5250
 130 GO SUB 5100
 140 GO SUB 5150
 150 GO SUB 5000
 160 GO SUB 5050
 200 REM ******INICIO**********
 210 GO SUB 4000
 220 GO SUB 4500
 230 GO SUB 6100
 240 GO SUB 5000
 250 GO SUB 5250
 260 GO SUB 5100
 270 GO SUB 5150
 280 GO SUB 5300
 290 GO SUB 6050
 300 GO SUB 5800
 310 GO SUB 5400
 320 GO SUB 6200
 330 GO SUB 5700
 340 GO SUB 5050
 350 GO TO 200
4000 REM *MOV.HORIZONTAL**
4010 IF INKEY$="5" AND ATTR (L+3
,C-1)<>2 AND ATTR (L,C-1)<>2 THE
N LET C=C-1: LET D=1: LET Z=Z+1:
 IF Z=4 THEN LET Z=0
4020 IF INKEY$="8" AND ATTR (L+3
,C+2)<>2 AND ATTR (L,C+2)<>2 THE
N LET C=C+1: LET D=5: LET Z=Z+1:
 IF Z=4 THEN LET Z=0
```

```
4030 RETURN
4500 REM **MOV.VERTICAL**
4510 IF INKEY$="7" THEN GO SUB 5
350
4520 RETURN
5000 REM ****PINTA HOMBRE*****
5010 FOR N=1 TO 4
5020 PRINT AT (N-1)+L,C; INK 7;B
$(D+Z,N)
5030 NEXT N
5040 RETURN
5050 REM ****BORRA HOMBRE***
5060 FOR N=1 TO 4
5070 PRINT AT (N-1)+L,C;"  "
5080 NEXT N
5090 RETURN
5100 REM ***PINTA ARANA******
5105 IF NP=2 OR NP=11 THEN GO TO
 5110
5108 RETURN
5110 IF NP=2 THEN PRINT AT 15,10
; INK 6;C$(INT (RND*3)+1)
5120 IF NP=11 THEN PRINT AT 15,5
; INK 6;C$(INT (RND*3)+1)
5130 RETURN
5150 REM ****PINTA NAVE******
5155 IF NP<>16 THEN RETURN
5160 PRINT INK 5;AT LN,9;"UXY"
5165 PRINT INK 5;AT LN+1,9;"Z[\"
5170 PRINT INK 5;AT LN+2,9;"]↑_"
5175 PRINT INK 5;AT LN+3,9;"£ab"
5180 RETURN
5250 REM ***PINTA TUERCA*****
5255 IF NP<>NPT THEN RETURN
5260 PRINT INK 3;AT LT,CT;"@"
5270 RETURN
5300 REM ***LOCALIZA TUERCA****
5305 IF NP<>NPT THEN RETURN
5310 IF ATTR (L+3,C)=3 OR ATTR (
L+3,C+1)=3 THEN LET NT=NT+1: LET
 NPT=20: BEEP 0.1,10: LET T=1: G
O SUB 5600
5320 RETURN
5350 REM ******VUELO*********
5355 IF ATTR (L,C-1)=2 OR ATTR (
L,C+2)=2 OR ATTR (L+3,C-1)=2 OR
ATTR (L+3,C+2)=2 THEN GO SUB 505
0: GO TO 5400
5358 IF ATTR (L-1,C)=2 OR ATTR (
L-1,C+1)=2 THEN GO SUB 5050: GO
TO 5400
5360 GO SUB 4000
5362 LET Z=1
5365 GO SUB 5850
5367 LET L=L-.5
5370 GO SUB 5000
5375 GO SUB 5100
5380 GO SUB 5150
5381 GO SUB 5250
5382 GO SUB 6050
5384 GO SUB 6100
5386 GO SUB 5800
5388 GO SUB 5050
5390 GO TO 5350
5400 REM *****CAIDA*********
5402 IF L=17 AND (NP=14 OR NP=15
) THEN RETURN
```





```
5405 IF ATTR (L+4,C)=2 OR ATTR (
L+4,C+1)=2 THEN RETURN
5406 GO SUB 5050
5407 GO SUB 6100
5409 GO SUB 4000
5410 LET L=INT (L+1)
5412 LET Z=1
5415 GO SUB 5850
5420 GO SUB 5000
5425 GO SUB 5250
5430 GO SUB 5100
5435 GO SUB 5150
5440 GO SUB 5300
5445 GO SUB 6050
5450 GO SUB 5600
5455 GO SUB 5700
5460 GO SUB 5050
5465 GO TO 5400
5500 REM **PINTA LOS HOMBRES****
5505 FOR N=22 TO 28 STEP 3
5508 BEEP .1,20
5510 FOR M=1 TO 4
5515 PRINT AT (M-1)*3,N; INK 4;B
$(6,M)
5520 NEXT M
5525 NEXT N
5530 RETURN
5550 REM **BORRA LOS HOMBRES***
5552 BEEP .1,20
5555 FOR N=30 TO 30-3*(3-NH) STE
P -1
5560 FOR M=1 TO 4
5565 PRINT AT (M-1)*3,N;"   "
5570 NEXT M
5575 NEXT N
5580 RETURN
5600 REM ***MUESTRA TUERCAS*****
5602 POKE 23607,60
5605 PRINT AT 10,25; INK 7;NT;
5608 POKE 23607,200: PRINT INK 6
;"0"
5610 IF NT=10 THEN PRINT AT 10,2
5; INK 7;"  "; FLASH 1;"0": BEEP
 0.1,20
5615 BEEP 0.1,10
5620 RETURN
5700 REM ***LLEGA A LA SALA****
5705 IF NP<>16 THEN RETURN
5708 IF T=0 THEN RETURN
5710 IF C<>10 THEN RETURN
5712 BEEP 0.1,10
5715 LET U=U+1: LET T=0
5720 PRINT AT 20,20+U; INK 5;"[redacted]"
5725 IF U=10 THEN PRINT AT 20,21
; INK 5; FLASH 1;"[redacted]": G
O SUB 6500
5740 GO SUB 5950
5750 RETURN
5800 REM **ACABA EL TIEMPO***
5805 IF TM<>0 THEN RETURN
5810 FOR N=1 TO 5: BEEP 1,20: NE
XT N
5815 FOR N=1 TO 20: PRINT AT N,0
; INK 2;"[redacted]"
5820 NEXT N: POKE 23607,60
5825 PRINT AT 10,4; INK 7;"Otra
partida?"
5830 IF INKEY$="S" OR INKEY$="s"
 THEN GO TO 50
5835 IF INKEY$="N" OR INKEY$="n"
 THEN STOP
5840 GO TO 5830
5850 REM **PATALLA SIGUIENTE V**
5852 IF (L=16 OR L=1) AND NP<>1
AND NP<>16 AND NP<>14 AND NP<>15
 THEN LET NP=NP+(SGN (15-L)): LE
T L=16-L+SGN (L-1): GO SUB 5900
5855 RETURN
5900 REM ****PINTA PANTALLA*****
5906 FOR n=0 TO 20: PRINT AT n,0
;"                         ": NEXT n
5907 FOR M=1 TO 4: FOR N=1 TO 4
5908 IF A$(NP,M,N)="3" THEN GO S
UB 5930
5909 IF A$(NP,M,N)="1" THEN GO S
UB 5913
5910 IF A$(NP,M,N)="2" THEN GO S
UB 5918
5911 NEXT N: NEXT M
5912 RETURN
5913 LET U=(M-1)*5
5914 LET V=(N-1)*5
5915 FOR J=1 TO 5
5916 PRINT INK 2;AT U+J,V;"PPPPP
"
5917 NEXT J: RETURN
5918 LET U=(M-1)*5
5919 LET V=(N-1)*5
5920 FOR J=2 TO 5
5921 PRINT INK 1;AT U+J,V;"[redacted]
"
5922 NEXT J: RETURN
5930 LET U=(M-1)*5
5935 LET V=(N-1)*5
5940 FOR J=1 TO 3: PRINT INK 2;A
T U+J,V;"PPPPP"
5945 NEXT J: RETURN
5950 REM ***VALOR TUERCA*****
5952 LET NPT=INT (RND*16)
5955 FOR N=1 TO 4
5960 IF A$(NP,N,2)="0" THEN GO T
O 5968
5965 NEXT N
5966 LET N=4
5968 LET LT=N*5: LET CT=9
5970 RETURN
6050 REM ********MATADO********
6051 LET MJ=0
6052 IF ATTR (L+3,C-1)=1 OR ATTR
 (L+3,C+2)=1 THEN LET MJ=1: GO T
O 6060
6053 IF ATTR (L+3,C)=6 THEN GO T
O 6060
6055 RETURN
6060 FOR J=1 TO 4
6065 POKE 23607,2+J
```

```
6070 GO SUB 5000: BEEP .1,20: NEXT J
6075 POKE 23607,200
6080 LET NH=NH-1: GO SUB 5550
6085 IF NH=0 THEN GO TO 5810
6086 GO SUB 5050
6087 IF MJ=1 THEN PRINT AT L+3,C; INK 1;"█"
6088 LET NP=1: LET C=12: LET L=12: GO SUB 5900
6090 RETURN
6100 REM ***CONTROL DEL TIEMPO**
6105 LET TIEMPO=INT ((65536*PEEK 23674+32768*PEEK 23673+PEEK 23672)/100)
6110 LET TH=TH-TIEMPO
6112 IF TH<0 THEN LET TH=0
6115 POKE 23607,60
6120 PRINT INK 5;AT 15,22;"T:"; INK 3;TH
6125 POKE 23607,200
6130 RETURN
6200 REM **LOCALIZA PANTALLA M**
6201 LET DT=1
6202 IF (NP>=5 AND NP<=8) OR (NP>=13 AND NP<=16) THEN LET DT=-1
6205 IF C=19 OR C=0 THEN LET NP=NP+SGN (C-15)*DT: LET C=19-C+(SGN (-15+C)): GO SUB 5900
6210 RETURN
6500 REM ***FINAL GUAY*******
6502 FOR N=5 TO 14: FOR M=1 TO 5
```




```
: PRINT AT M,N;" ": NEXT M: NEXT N
6510 FOR N=11 TO 0 STEP -1
6520 LET LN=N: GO SUB 5150
6530 PRINT AT LN+4,9;"      "
6535 NEXT N
6540 CLS
6545 FOR N=1 TO 100: PLOT INK 7;RND*255,RND*175: NEXT N
6555 FOR N=0 TO 17
6560 LET LN=N: GO SUB 5150
6570 PRINT AT LN-1,9;"      "
6575 NEXT N
6578 LET L=17
6580 FOR C=13 TO 20
6585 LET D=5: LET Z=Z+1: IF Z=4 THEN LET Z=0
6590 GO SUB 5000
6595 PAUSE 10: GO SUB 5050
6600 NEXT C
6605 POKE 23607,60: PRINT AT 10,3; INK 7;"ENHORABUENA TE HAS SALVADO
                 OTRA?"
6608 POKE 23607,200
6610 LET Z=Z+1: IF Z=4 THEN LET Z=0
6615 GO SUB 5000
6620 IF INKEY$="S" OR INKEY$="s" THEN CLS : GO TO 50
6625 IF INKEY$="N" OR INKEY$="n" THEN STOP
6630 GO TO 6610
6900 REM ****INSTRUCIONES******
6910 POKE 23607,60: INK 0
6920 PRINT ''''''"SALVATE DEL PLANETA QUE EXPLOTA-RA CUANDO ACABE EL TIEMPO CON TUNAVE A LA QUE DEVERAS LLEVAR 10 TUERCAS PARA QUE DESPEGUE"''''''"USA:"''"5 Y 8  DERECHA E IZQUIERDA"''"7        VOLAR"
6925 PRINT ''"PULSA UNA TECLA"
6930 FOR N=22710 TO 23294: POKE N,RND*6+1
6935 IF INKEY$<>"" THEN RETURN
6940 NEXT N: GO TO 6930
7000 REM *****cloacas*****
7001 RESTORE 7010
7002 FOR n=1 TO 16
7003 FOR m=1 TO 4
7004 READ a$(n,m)
7005 NEXT m: NEXT n
7008 RETURN
7010 DATA "1111","1001","1000","1211","1131","0000","0101","1111","1111","0003","1000","1121","1101","3001","0011","1211","1111","0001","1101","1101 ","1111","1000","0001","1121","1111","1003","0000","1121","1011","1001","1100","1111"
7015 DATA "1111","1000","1001","1011","1111","0000","1001","1121","1311","0000","1011","1111","3011","0011","1011","1111","1111","0011","1011","1011","1111","1100","1001","0011","1111","1011","0011","1000","1111","1001","1000","1111"
8000 REM ****CARACTERES*********
8005 RESTORE 9000
8010 FOR N=0 TO 760
8020 POKE 51456+N,PEEK (15616+N)
8030 NEXT N
8040 FOR N=51456 TO 51991
8050 READ A: POKE N,A
8060 NEXT N
8070 RETURN
9000 DATA BIN 11,BIN 100,BIN 110,BIN 100,BIN 10,BIN 1,BIN 1,BIN 1
9010 DATA BIN 10000000,BIN 1000000,BIN 1000000,BIN 1000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 11000000
9020 DATA BIN 11,BIN 11,BIN 11,BIN 110,BIN 1110,BIN 111100,0,BIN 11
9030 DATA BIN 1100000,BIN 100000,BIN 1100000,BIN 110000,BIN 1110000,BIN 1100000,0,BIN 11000000
9040 DATA BIN 11,BIN 11,BIN 11,BIN 101,BIN 101,BIN 101,BIN 1110,BIN 1110
9050 DATA BIN 11000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 11000000,BIN 11000000,BIN 11000000,BIN 11100000
9060 DATA BIN 1100,BIN 1100,BIN 110,BIN 110,BIN 110,BIN 11,BIN 11,BIN 1110
9070 DATA BIN 1100000,BIN 1110000,BIN 110000,BIN 11000,BIN 11100,BIN 1100,BIN 1110,BIN 101110
9080 DATA BIN 11,BIN 100,BIN 100,BIN 100,BIN 10,BIN 11,BIN 11,BIN 111
9090 DATA BIN 10000000,BIN 1000000,BIN 11000000,BIN 1000000,BIN 10000000,0,0,0
9100 DATA BIN 1101,BIN 1101,BIN 1101,BIN 11000,BIN 11100,BIN 1100,0,BIN 111
9120 DATA BIN 10000000,BIN 10000000,BIN 10000000,BIN 11000000,BIN 11100000,BIN 1111000,0,BIN 10000000
9130 DATA BIN 111,BIN 11,BIN 11,BIN 11,BIN 111,BIN 111,BIN 110,BIN 1110
9140 DATA BIN 10000000,BIN 10000000,BIN 10000000,BIN 1000000,BIN 1000000,BIN 1000000,BIN 11100000,BIN 11100000
9150 DATA BIN 1100,BIN 11100,BIN 11000,BIN 110000,BIN 1110000,BIN 1100001,BIN 11100001,BIN 10111000
9160 DATA BIN 1100000,BIN 1100000,BIN 11000000,BIN 11000000,BIN 11000000,BIN 10000000,BIN 10000000,BIN 11100000
9170 DATA BIN 11,BIN 100,BIN 100,BIN 100,BIN 10,BIN 11,BIN 101,BIN 101
9180 DATA BIN 10000000,BIN 1000000,BIN 11000000,BIN 1000000,BIN 10000000,0,0,BIN 10000000
9190 DATA BIN 101,BIN 101,BIN 101,BIN 101,0,BIN 100,BIN 100,BIN 100
9200 DATA BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 11000000,BIN 11000000,BIN 11000000,BIN 1100000
9210 DATA BIN 111,1,1,1,2,2,2,2
9220 DATA BIN 110000,BIN 11000000,BIN 11000000,BIN 11000000,BIN 11000000,BIN 11000000,BIN 11100000,BIN 1100000
9230 DATA 2,1,1,1,BIN 11,1,BIN 110,BIN 1011
9240 DATA BIN 11100000,BIN 11000000,BIN 11000000,BIN 10000000,BIN 10000000,0,0,BIN 11000000
9250 DATA BIN 11,BIN 100,BIN 110,BIN 100,2,1,1,BIN 11
9260 DATA BIN 10000000,BIN 1000000,BIN 1000000,BIN 1000000,BIN 10000000,BIN 10000000,BIN 1000000,BIN 1000000
9270 DATA BIN 11,BIN 11,BIN 11,BIN 11,BIN 110,BIN 110,BIN 110,BIN 1100
9280 DATA BIN 1000000,BIN 1000000,BIN 1000000,BIN 1000000,0,BIN 1000000,BIN 1000000,BIN 1000000
9290 DATA BIN 11001,BIN 111,BIN 111,BIN 111,BIN 110,BIN 110,BIN 1110,BIN 1100
9300 DATA BIN 11000000,0,0,0,BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000
9310 DATA BIN 1110,BIN 111,BIN 111,BIN 11,BIN 11,1,0,BIN 111
9320 DATA BIN 10000000,0,0,0,BIN 10000000,0,BIN 11000000,BIN 10000000
9330 DATA BIN 11,BIN 100,BIN 100,BIN 100,2,BIN 11,BIN 11,BIN 111
9340 DATA BIN 10000000,BIN 1000000,BIN 11000000,BIN 1000000,BIN 10000000,0,0,0
9350 DATA BIN 110,BIN 110,BIN 110,BIN 1110,BIN 1100,BIN 11000,BIN 11100,BIN 1110,0
9360 DATA BIN 10000000,BIN 10000000,BIN 10000000,BIN 11000000,BIN 11100000,BIN 11111000,BIN 10000000,0
9370 DATA BIN 111,BIN 11,BIN 11,1,BIN 100,BIN 110,BIN 110,BIN 1110
9380 DATA BIN 10000000,BIN 10000000,BIN 10000000,BIN 11000000,BIN 11000000,BIN 11000000,BIN 11100000,BIN 1100000
9390 DATA BIN 1110,BIN 1100,BIN 11100,BIN 11000,BIN 110000,BIN 1110000,BIN 11100000,BIN 10111000
9400 DATA BIN 1110000,BIN 110000,BIN 1100000,BIN 1100000,BIN 11000000,BIN 11000000,BIN 11000000,BIN 10110000
9410 DATA BIN 11,BIN 100,BIN 110,BIN 100,2,1,1,1
9420 DATA BIN 11000000,BIN 1000000,BIN 1000000,BIN 1000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 11000000
9430 DATA BIN 10,BIN 110,BIN 110,BIN 1110,BIN 11110,BIN 1111110,2,0
9440 DATA BIN 11000000,BIN 11000000,BIN 11100000,BIN 1100000,BIN 110000,BIN 1110000,BIN 11100000,0
9450 DATA BIN 11,BIN 11,BIN 11,BIN 111,BIN 110,BIN 110,BIN 1110,BIN 1100
9460 DATA BIN 11000000,BIN 10000000,BIN 10000000,0,BIN 1000000,BIN 11000000,BIN 11000000,BIN 11100000
9470 DATA BIN 11100,BIN 10000,BIN 1100,BIN 1100,BIN 110,BIN 110,BIN 110,BIN 11010
9480 DATA BIN 11100000,BIN 1100000,BIN 1100000,BIN 110000,BIN 11000,BIN 1110,BIN 1110,BIN 111010
9490 DATA BIN 11110111,BIN 11110111,BIN 11110111,0,BIN 11111110,BIN 11111110,BIN 11111110,0
9500 DATA BIN 111100,BIN 1111110,BIN 11111111,BIN 11100111,BIN 11100111,BIN 11111111,BIN 1111110,BIN 111100
9510 DATA 0,0,0,0,0,0,0,0
9600 DATA BIN 100,BIN 1010,BIN 10001,BIN 100011,BIN 1000100,BIN 1001000,BIN 1000100,BIN 100010
9605 DATA BIN 1111100,BIN 11111110,BIN 10110111,BIN 10010011,BIN 11111110,BIN 1111100,0,0
9610 DATA BIN 1000000,BIN 10100000,BIN 10000,BIN 10001000,BIN 1000100,BIN 100100,BIN 1000100,BIN 10001000
9615 DATA BIN 1111100,BIN 11111110,BIN 11011011,BIN 10010011,BIN 11111110,BIN 1111100,0,0
9620 DATA 0,0,0,0,0,0,1,1
9625 DATA BIN 11000,BIN 100100,BIN 100100,BIN 100100,BIN 1000010,BIN 10011001,BIN 100100,BIN 1011010
9630 DATA 0,0,0,0,0,0,BIN 10000000,BIN 10000000
9635 DATA BIN 1,1,1,1,1,1,1,1
9640 DATA BIN 10100101,BIN 1000010,BIN 10000001,BIN 111100,BIN 1000010,BIN 1000010,BIN 1000010,BIN 1000010
9645 DATA BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000
9650 DATA BIN 1,1,1,1,1,1,1,1
9655 DATA BIN 1000010,BIN 1000010,BIN 1000010,BIN 1000010,BIN 1000010,BIN 11000011,BIN 11000011,BIN 11000011
9670 DATA BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000,BIN 10000000
9675 DATA BIN 1,BIN 110,BIN 1001,BIN 10001,BIN 100000,BIN 1000000,BIN 10000000,BIN 11111111
9680 DATA 0,BIN 11000,BIN 100100,BIN 1000010,BIN 10000001,BIN 10000001,BIN 10000001,BIN 11111111
9685 DATA BIN 10000000,BIN 1100000,BIN 10010000,BIN 10001000,BIN 100,BIN 10,1,BIN 11111111
```

Profesor particular



Arturo LOBO y J. J. LEON

Hace unos cuantos días que ha comenzado el curso escolar... y se nos ha ocurrido que una buena parte de aquéllos que inician esta terrible peripecia verían con buenos ojos la ayuda «desinteresada» de un profesor particular que les echara una mano con los deberes. Aquí lo tenéis.

REPRESENTACION TRIDIMENSIONAL

Este programa permite representar cualquier tipo de superficie y las curvas planas o alabeadas. Dibuja en pantalla el entorno elegido de la superficie, introducida mediante sus ecuaciones.

El «input» de las ecuaciones está pensado para ofrecer la mayor comodidad posible. Estas pueden ser introducidas en coordenadas (x,y,z), cilíndricas (r,θ,z) o esféricas (r,θ,φ), tanto en forma paramétrica —funciones de t y s— como explícita —una función de las otras dos—. Sin embargo, cuando se quiere representar una curva sólo está permitida la forma paramétrica de las ecuaciones.

Después, el programa pide entre qué límites la queremos representar y desde qué visual (vx, vy, vz), que representa la dirección desde la que miramos.

Primero representa la posición de la superficie respecto a los ejes y luego hace una ampliación de ésta. El resultado es francamente espectacular, como podréis comprobar.

Determinadas superficies, contempladas desde ciertas vistas, pueden resultar poco claras; sólo tendréis que cambiar la visual adecuadamente para que mejore notablemente la apreciación.

Se debe recordar que una superficie es cualquier función z=f(x,y) o cualesquiera funciones x=f(t,s), y=g(t,s), z=h(t,s), así como que las curvas sólo pueden tener un parámetro.

Ahora os damos unos cuantos ejemplos:

Esfera: r=2 y ENTER, θ=t y ENTER, φ=s y ENTER, tO=O, t1=2 PI, sO=−PI/2, s1=PI/2, (vx,vy,vz)=(1,1,1).

Helicoide recto: r=t, θ=s, z=s/2 con tO=1, t1=4, θO=O, 1=7 y (vx, vy,vz)=(1,1,1).

Probad también la siguiente función y fijaos bien en su estructura:

z=SQR(1−x•x+y•y) • (x•x+y•y<1) y 2 veces ENTER con xO=−2, x1=2, yO=−2, y1=2.

Recordad que para hacer x^2 en el Spectrum debéis poner ABSx↑2, pues no admite potencias de números negativos.

La variable φ se escribe en modo gráfico y A y la variable θ en modo gráfico y B.

```
5999 GO TO 7000
6000 DEF FN x(t,s)=VAL x$: DEF FN y(t,s)=VAL y$: DEF FN z(t,s)=VAL z$: BORDER 1: PAPER 1: INK 7: CLS : PRINT AT 20,11; FLASH 1;"CALCULANDO"
6500 LET ilm=SQR (ABS vx↑2+ABS vy↑2): LET jlm=ilm*SQR (ABS vx↑2+ABS vy↑2+vz*vz): LET ax=-vy/ilm: LET ay=vx/ilm: LET bx=-vx*vz/jlm: LET by=-vy*vz/jlm: LET bz=(vx*vx+vy*vy)/jlm
6510 DEF FN p(t,s)=ax*FN x(t,s)+ay*FN y(t,s)
6513 DEF FN q(t,s)=bx*FN x(t,s)+by*FN y(t,s)+bz*FN z(t,s)
6516 DEF FN s(x,y,z)=ax*x+ay*y
6519 DEF FN r(x,y,z)=bx*x+by*y+bz*z
6525 LET xm=SQR (ax*ax+bx*bx): LET ym=SQR (ay*ay+by*by): LET zm=ABS bz
6528 DIM a(2,20,20): LET xt=FN p(t0,s0): LET yt=FN q(t0,s0): LET lm=xt*xt+yt*yt: LET xa=xt: LET xi=xt: LET ya=yt: LET yi=yt
6531 FOR y=1 TO 7: LET s=s0+(y-1)/6*(s1-s0): FOR x=1 TO 7: LET t=t0+(t1-t0)*(x-1)/6: LET xt=FN p(t,s): LET yt=FN q(t,s): LET a(1,y,x)=xt: LET a(2,y,x)=yt: LET lm=lm+(xt*xt+yt*yt-lm)*(xt*xt+yt*yt>lm): LET xa=xa+(xt-xa)*(xt>xa): LET xi=xi+(xt-xi)*(xi>xt): LET ya=ya+(yt-ya)*(yt>ya): LET yi=yi+(yt-yi)*(yi>yt): NEXT x: NEXT y: LET lm=SQR lm
6540 LET dm=ABS (xa-xi)*.7: LET dm=dm+(ABS (ya-yi)-dm)*(ABS (ya-yi)>dm)
6543 LET dmax=dm*1.36: LET dmax=dmax+(lm-dmax)*(lm>dmax)
6546 LET cb=87/dmax
6549 LET x1x=FN s(dmax/xm,0,0)*cb: LET x1y=FN r(dmax/xm,0,0)*cb: LET y1x=FN s(0,dmax/ym,0)*cb: LET y1y=FN r(0,dmax/ym,0)*cb: LET z1x=FN s(0,0,dmax/zm)*cb: LET z1y=FN r(0,0,dmax/zm)*cb
6552 CLS : PLOT 127,87: DRAW x1x,x1y: PLOT 127,87: DRAW y1x,y1y: PLOT 127,87: DRAW z1x,z1y: PRINT AT INT (11-x1y/8),INT (17+x1x/8);"X": PRINT AT INT (11-y1y/8),INT (17+y1x/8);"Y": PRINT AT INT (11-z1y/8),INT (17+z1x/8);"Z"
6554 LET ps=6: LET pt=6: LET cx=127: LET cy=87: LET cb2=cb: GO SUB 6560
6559 GO TO 6568
6560 LET kt=pt+1: FOR s=1 TO (ps+1): POKE (52368+kt*s),a(1,s,1)*cb2+cx: POKE (52868+kt*s),a(2,s,1)*cb2+cy: PLOT PEEK (52368+kt*s),PEEK (52868+kt*s): FOR t=2 TO (pt+1): POKE (52367+kt*s+t),a(1,s,t)*cb2+cx: POKE (52867+kt*s+t),a(2,s,t)*cb2+cy: DRAW PEEK (52367+kt*s+t)-PEEK (52366+kt*s+t),PEEK (52867+kt*s+t)-PEEK (52866+kt*s+t): NEXT t: NEXT s
6565 FOR t=1 TO (pt+1): PLOT PEEK (52367+kt+t),PEEK (52867+kt+t): FOR s=2 TO (ps+1): DRAW PEEK (52367+kt*s+t)-PEEK (52367+kt*(s-1)+t),PEEK (52867+kt*s+t)-PEEK (52867+kt*(s-1)+t): NEXT s: NEXT t
6567 RETURN
6600 PRINT #0;TAB 11; FLASH 1;"CALCULANDO": LET PO2=52375: LET PO1=52917: LET PO3=52381: GO SUB 6630: LET Area=Ar: LET PO=52923: GO SUB 6630: LET Area=Area+Ar: LET Area=(2*Area+3000)/3
6629 GO TO 6632
6630 LET ar=ABS ((PEEK po3-PEEK po2)*(PEEK (po1+500)-PEEK (po2+500))-(PEEK po1-PEEK po2)*(PEEK (po3+500)-PEEK (po2+500)))/2: RETURN
6635 LET xmd=(xa+xi)/2: LET ymd=(ya+yi)/2: LET cb2=200/dmax: LET cx=127-xmd*cb2: LET cy=87-ymd*cb2: LET ncu=Area*5.3/10↑2
6638 LET lt=1: FOR s=1 TO 6: LET po1=52367+7*s+4: LET po2=52367+7*(s+1)+4: GO SUB 6650: LET lt=lt+lo: NEXT s: LET ls=1: FOR t=1 TO 6: LET po1=52367+7*4+t: LET po2=52367+7*4+(t+1): GO SUB 6650: LET ls=ls+lo: NEXT t
6649 GO TO 6651
6650 LET lo=SQR ((ABS (PEEK PO1-PEEK PO2))↑2+(ABS (PEEK (PO1+500)-PEEK (PO2+500)))↑2): RETURN
6660 LET pt=SQR (ncu*lt/ls): LET ps=ncu/pt: LET pt=INT (pt-(pt-19)*(19<pt)+(6-pt)*(pt<6)): LET ps=INT (ps-(ps-19)*(19<ps)+(6-ps)*(ps<6))
6665 LET acx=pt: LET pt=ps: LET ps=acx
6700 FOR y=1 TO (ps+1): LET s=s0+(y-1)/(ps+(ps=0))*(s1-s0): FOR x=1 TO (pt+1): LET t=t0+(x-1)/(pt+(pt=0))*(t1-t0): LET A(1,y,x)=FN p(t,s): LET A(2,y,x)=FN q(t,s): NEXT x: NEXT y
6705 CLS
6710 GO SUB 6560: GO TO 7180
7000 INK 0: LET t0=0: LET t1=1: LET s0=0: LET s1=1: FOR i=USR "a" TO USR "a"+15
7010 READ pk: POKE i,pk: NEXT i
7020 DATA 0,36,74,82,60,16,16,8,0,60,66,66,126,66,60,0
7030 DIM m$(3,3): DIM n$(3,3): FOR i=1 TO 3: FOR j=1 TO 3: READ a$: LET m$(i,j)=a$: READ a$: LET n$(i,j)=a$: NEXT j: NEXT i
7040 DATA "x","x","y","y","z","z","r","r"," ","g","z","z","r","r"," ","g"," ","f"
7050 PAPER 6: BORDER 6: CLS : PRINT AT 4,4;"REPRESENTACION ESPACIAL";AT 6,4;"DE CURVAS Y SUPERFICIES"
7060 PRINT AT 18,0;"INTRODUCE LA/S ECUACION/ES:"
7063 PRINT AT 1,0;"(  y   son A,B en modo grafico)"
7065 LET j$="": LET k$=""
7070 INPUT LINE i$
7075 INPUT LINE j$: IF j$="" THEN LET ecs=1: GO TO 7100
7080 INPUT LINE k$: IF k$="" THEN LET ecs=2: GO TO 7100
7090 LET ecs=3
7100 LET l$=i$+j$+k$: GO SUB 8050
7110 LET t=0: LET s=0: GO SUB 8100
7115 PRINT AT 21,0;"Mete los limites:"
7120 IF ts=0 AND ecs=1 THEN GO SUB 8200: GO TO 7160
7130 IF ecs<>3 OR ts=0 THEN RUN 7000
7140 IF t=1 THEN GO SUB 8400
7150 IF s=1 THEN GO SUB 8500
7151 FOR i=1 TO LEN l$: IF CODE l$(i)=144 OR CODE l$(i)=145 THEN LET l$(i)=CHR$ (CODE l$(i)-42)
7152 NEXT i: LET i$=l$( TO LEN i$): LET j$=l$(LEN i$+1 TO LEN i$+LEN j$): LET k$=l$(LEN i$+LEN j$+1 TO )
7153 LET pk=(PEEK 23627+256*PEEK 23628)-24: POKE pk+7,116: POKE pk+15,115
7155 POKE pk,CODE i$(1): LET i$=i$(3 TO ): GO SUB 9999: POKE pk,CODE j$(1): LET i$=j$(3 TO ): GO SUB 9999: POKE pk,CODE k$(1): LET i$=k$(3 TO ): GO SUB 9999
7160 IF tipo=2 THEN GO SUB 8030
7170 IF tipo=3 THEN GO SUB 8000
7180 INPUT "(vx,vy,vz)?"'vx'vy'vz: GO TO 6000
8000 LET x$="("+r$+")*COS ("+f$+")*COS ("+g$+")"
8010 LET y$="("+r$+")*COS ("+f$+")*SIN ("+g$+")"
8020 LET z$="("+r$+")*SIN ("+f$+")": RETURN
8030 LET x$="("+r$+")*COS ("+g$+")"
8040 LET y$="("+r$+")*SIN ("+g$+")": RETURN
8050 FOR i=1 TO LEN l$: IF CODE l$(i)=144 THEN LET tipo=3: RETURN
8055 NEXT i
8060 FOR i=1 TO LEN l$: IF l$(i)="r" OR CODE l$(i)=145 THEN LET tipo=2: RETURN
8065 NEXT i: LET tipo=1: RETURN
8100 FOR i=1 TO LEN l$: IF l$(i)="t" THEN LET t=1
8110 IF l$(i)="s" THEN LET s=1
8120 NEXT i: LET ts=t+s: RETURN
8200 LET pk=(PEEK 23627+256*PEEK 23628)-25: FOR i=1 TO 3: IF i$(1)<>m$(tipo,i) THEN LET pk=pk+8: POKE pk,CODE n$(tipo,i): GO TO 8220
8210 LET ind=i
8220 NEXT i: GO SUB 9999
8230 FOR i=1 TO LEN i$: FOR j=1 TO 3: IF i$(i)=m$(tipo,j) AND j<>ind THEN LET i$(i)=VAL$ (n$(tipo,j)+"$")
8240 NEXT j: NEXT i: POKE pk-15,CODE n$(tipo,ind): LET i$=i$(3 TO ): GO SUB 9999
8250 LET i=1: IF i=ind THEN LET i=2
8260 PRINT #0;m$(tipo,i);"0? ",m$(tipo,i);"1? ": PAUSE 0: INPUT t0,t1: LET i=i+1
8270 IF i=ind THEN LET i=i+1
8280 PRINT #0;m$(tipo,i);"0? ",m$(tipo,i);"1? ": PAUSE 0: INPUT s0,s1: RETURN
8400 INPUT "t0 ";t0,"t1 ";t1: RETURN
8500 INPUT "s0 ";s0,"s1 ";s1: RETURN
9999 LET g$=i$: LET t$="t": LET s$="s": RETURN
```

Fighting Warrior

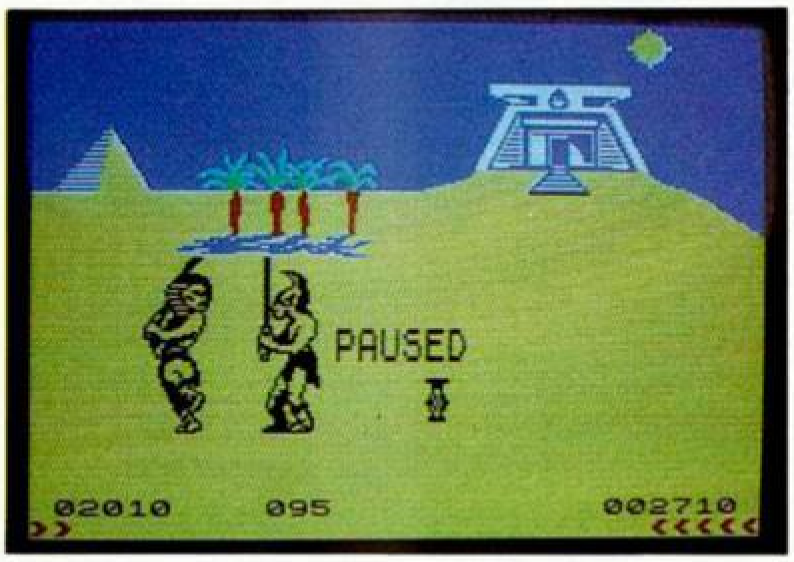
EL HEROE DEL ANTIGUO EGIPTO

Muy en la línea de Exploding Fist, Melbourne House ha creado este magnífico Fighting Warrior, pero esta vez ambientado en el mítico Egipto, cambiando las págodas por pirámides y los puños por espadas.

Melbourne House

Acción

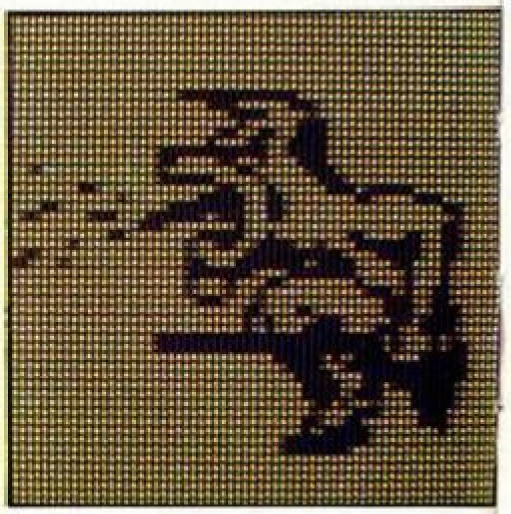
La vida transcurria con normalidad en aquella pequeña aldea egipcia. Sus habitantes vivian tranquilos dedicados alegremente a sus quehaceres contidianos. Pero toda esta armonía y felicidad se vió súbitamente truncada por la repentina aparición de unos seres monstruosos con cuerpo de hombre y cabeza de bestia feroz, que tras mantener una encarnizada lucha con los guerreros de la aldea, consiguieron raptar a la bella princesa y llevársela como prisionera a su Gran Pirámide.



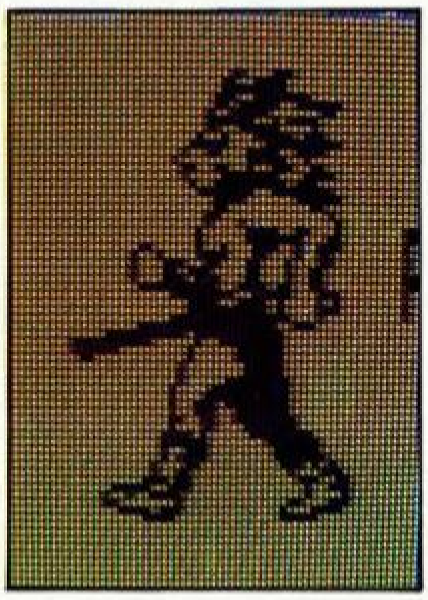
Fighting Warrior, miembro del poblado y el más valiente y aguerrido guerrero de entre todo el ejército del faraón, ha decidido ir en busca de su amada princesa. ¿Te gustaria acompañarle en tan noble misión? ¿Estarias dispuesto a enfrentarte a esos terribles monstruos para que volviera a reinar la paz en las lejanas tierras de Egipto?

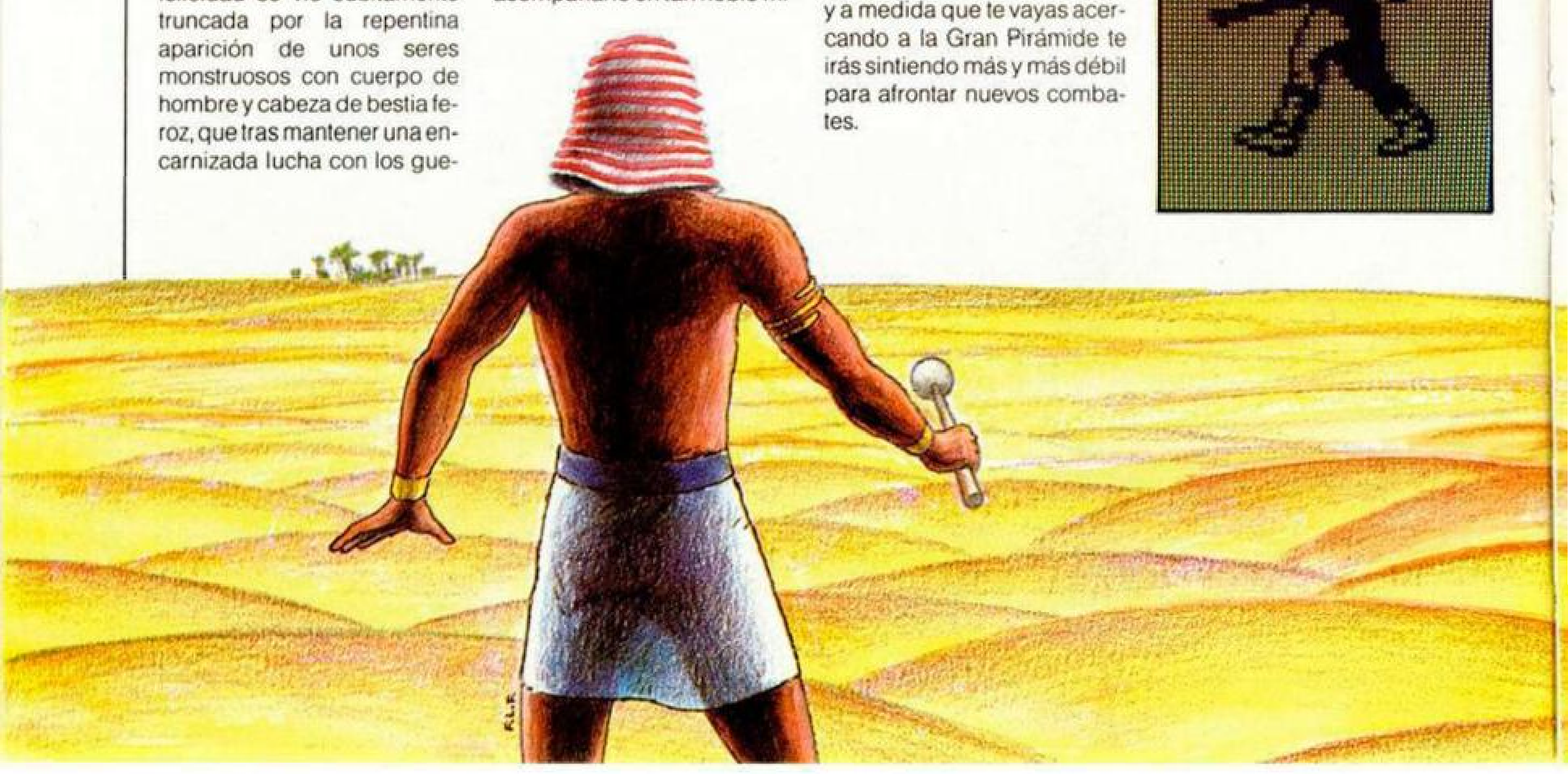
Antes de que tomes una decisión definitiva nos vemos en la obligación moral de advertite que muchos serán los peligros con los que tendrás que enfrentarte, pues las horribles bestias que saldrán a tu paso y ante las cuales deberás demostrar tu destreza con la espada, no son precisamente monjitas de la caridad, e intentarán evitar con todos los medios a su alcance que rescates a su prisionera.

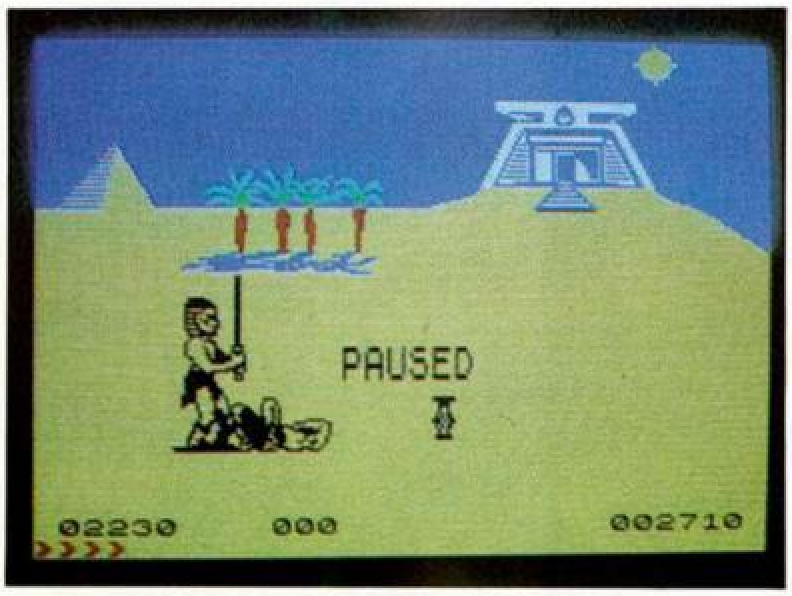
Además, como ya sabes, el clima del desierto es abrasador y el camino por recorrer es largo y lleno de sorpresas. Todo esto unido a las múltiples heridas que inevitablemente recibirás en los duros combates frente a tus enemigos, irán haciendo mella en tí, y a medida que te vayas acercando a la Gran Pirámide te irás sintiendo más y más débil para afrontar nuevos combates.

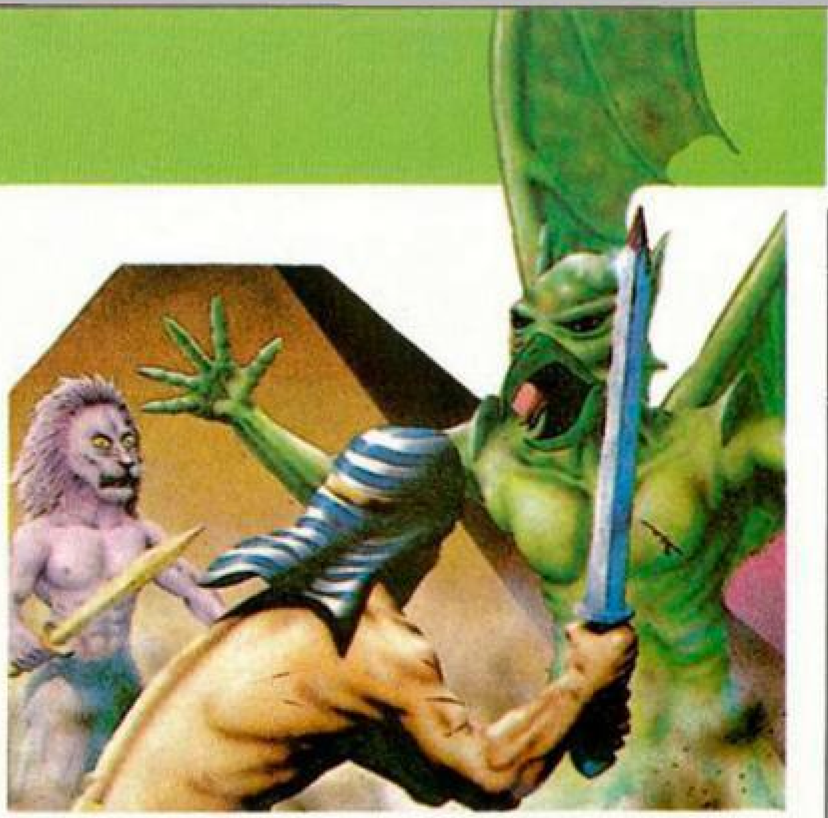


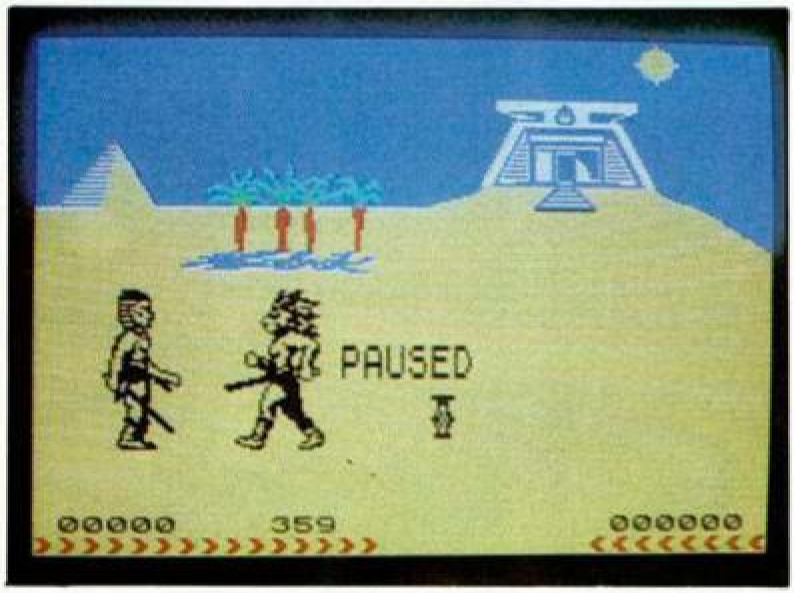


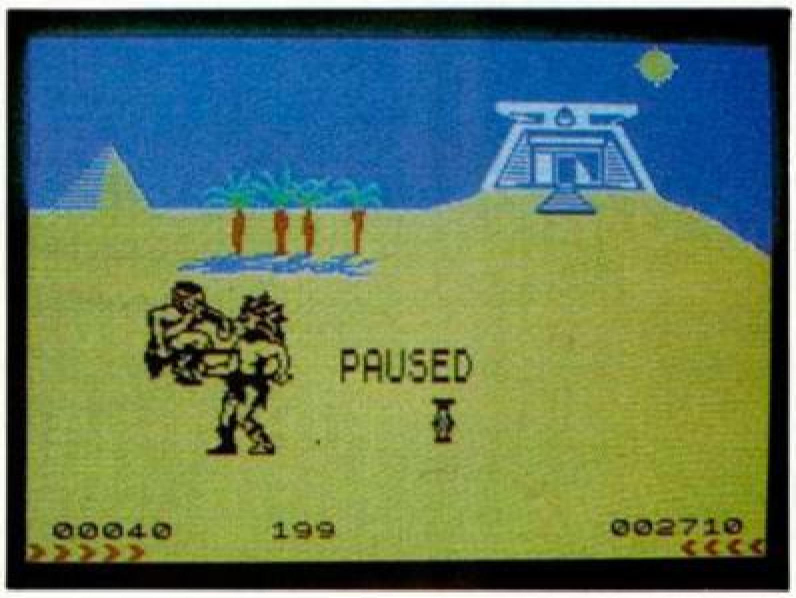






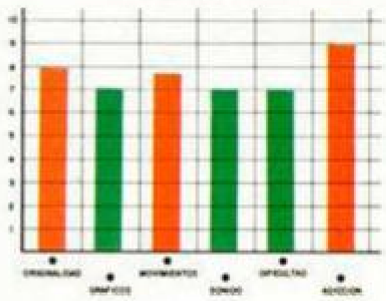


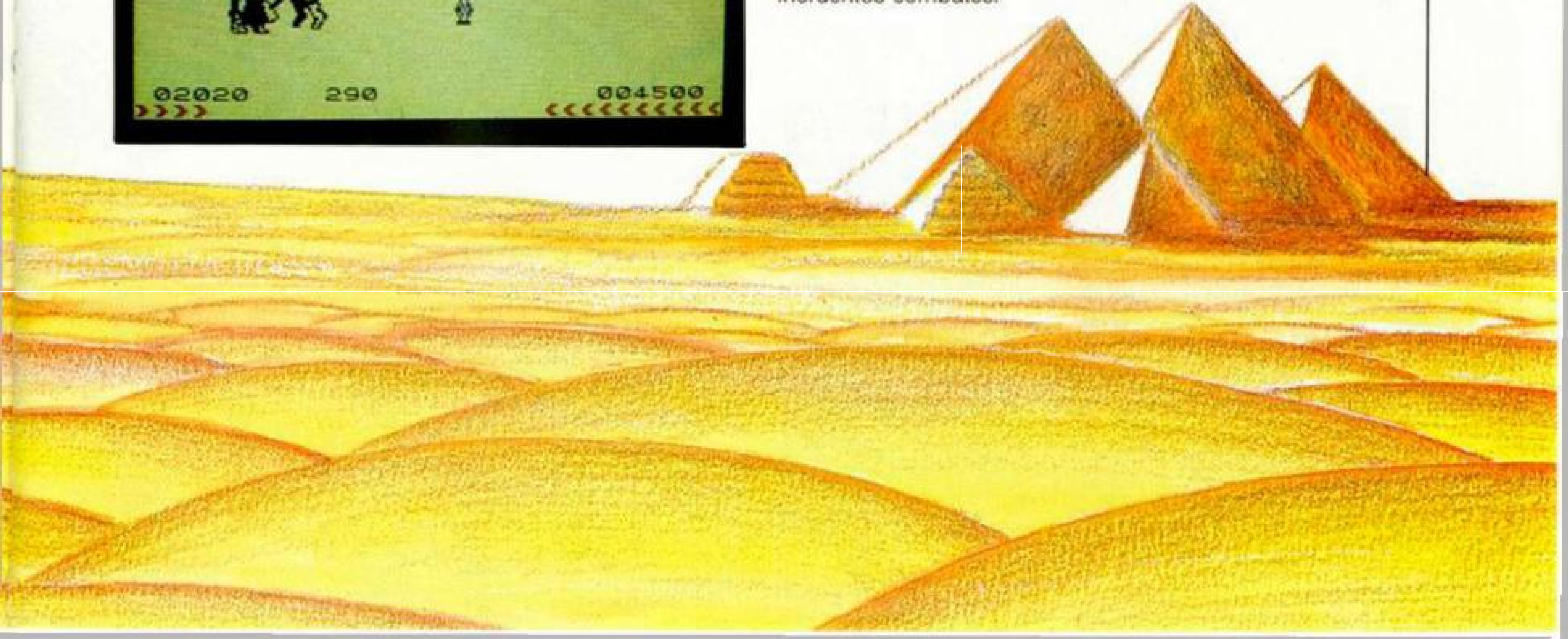






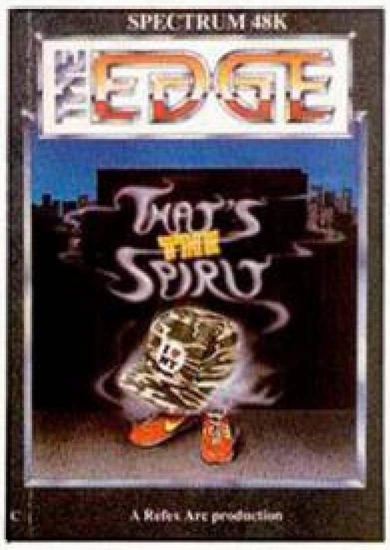
Si a pesar de todos estos inconvenientes decides acompañar a Fighting Warrior en tan peligrosa aventura, te aseguramos que te verás recompensado enórmemente en emoción y diversión, y si consigues rescatar a la princesa, seguro que querrás que la vuelvan a raptar para poder seguir disfrutando con estos incruentos combates.

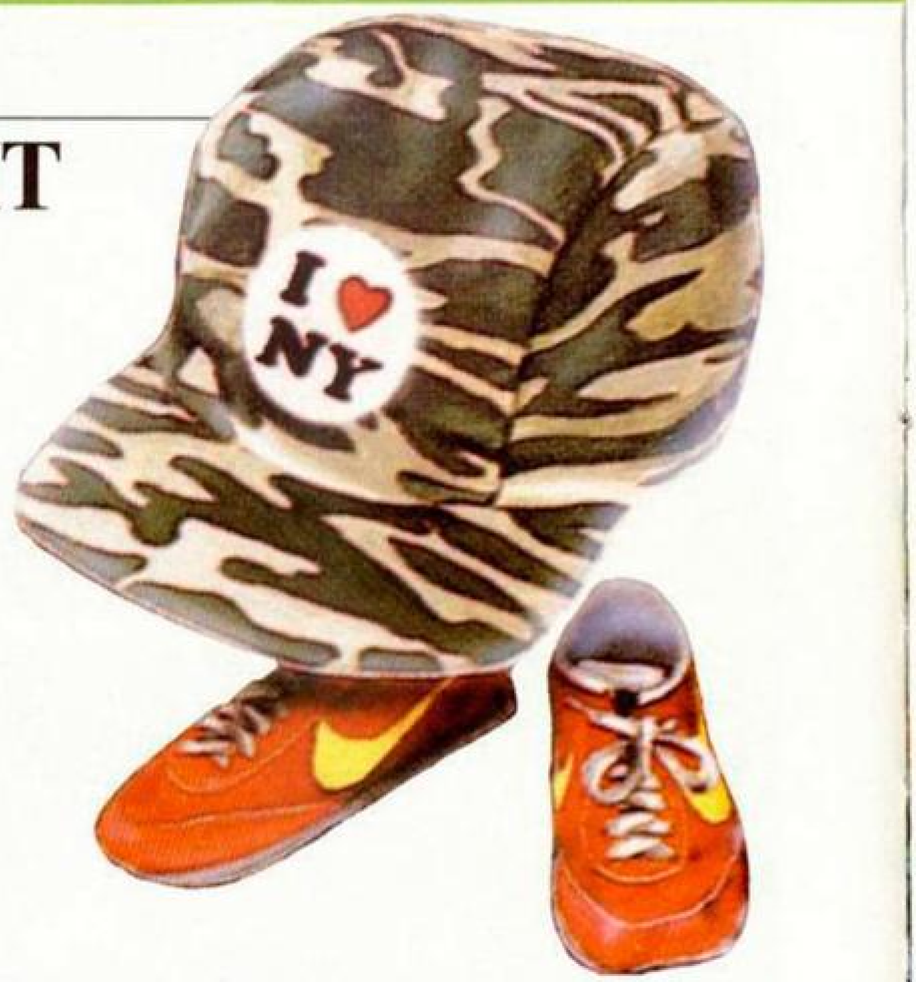




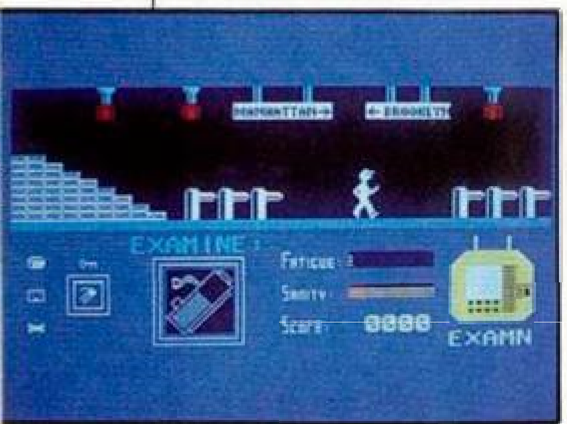
A la caza de fantasmas

THAT'S THE SPIRIT



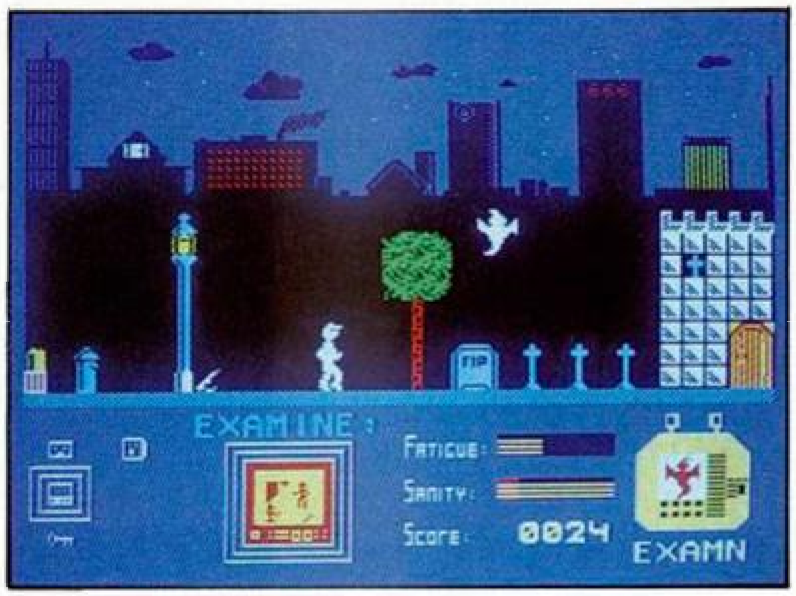


The edge
Aventura gráfica



«Tiempo para completarlo: meses, incluso para los más experimentados jugadores de aventuras», advierte la carátula de este juego. Y creo que no exagera en absoluto. Nos encontramos frente a uno de los juegos más complicados y de difícil solución de los exitentes hasta ahora en el mercado, el That's the Spirit. Se trata de una especie de aventura conversacional, pero sin texto, en la que debemos «dialogar» con el personaje, a través de una serie de acciones ya establecidas en el teclado tales como coger, soltar, salir, dormir, comer, entrar, y así hasta un total de dieciséis, mediante las cuales se nos permite operar con los diferentes objetos que encontramos en nuestro camino y que nos servirán para lograr nuestro fin: capturar a todos los espíritus que vagabundean por la ciudad de New York.

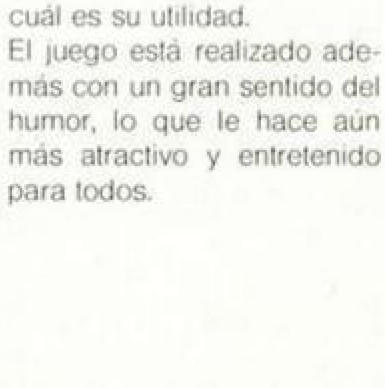
A simple vista podría parecer que tampoco es para tanto y que tarde o temprano, realizando pruebas con los objetos, iremos consiguiendo que las cosas se nos pongan cada vez más favorables y que poco a poco iremos acabando con nuestros etéreos enemigos. Pues desde ya, tened muy claro que no os va a resultar nada, pero que nada fácil, y que incluso en muchas ocasiones desesperaréis en los múltiples intentos de sacarle algún provecho a los utensilios variados que hayáis ido recolectando en vuestro árduo caminar.—¡Imposible!— exclamaréis, —¡Oh dioses, dadme una señal que me indique qué es lo que debo hacer!— clamaréis a los cuatro vientos. Pero todo será en vano. Sólo grandes dosis de paciencia y el quedaros muchos días sin cenar, os permitirán obtener algún que otro resultado positivo, que os dará fuerzas para continuar con esta misión, os aseguro que por lo menos para nosotros es casi imposible.



Para terminar os diremos que los gráficos son realmente maravillosos, aunque presenta el problema de que en algunas ocasiones resulta bastante difícil adivinar qué es el objeto que hemos cogido, y cuál es su utilidad.

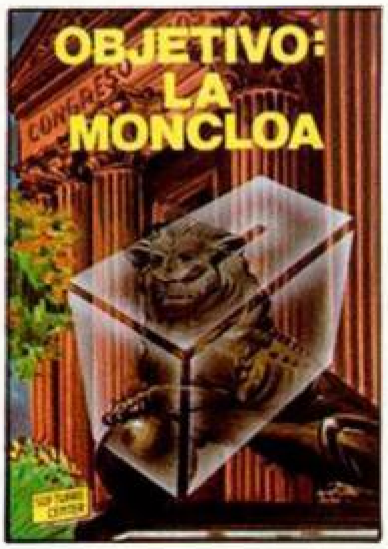
El juego está realizado además con un gran sentido del humor, lo que le hace aún más atractivo y entretenido para todos.





Rige el destino de España

OBJETIVO LA MONCLOA



Software Center
Estrategia

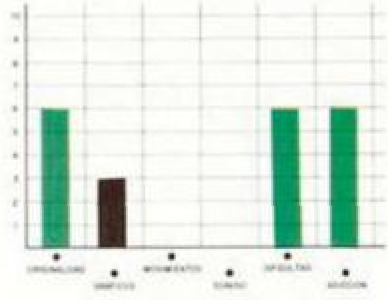
Software Center nos ofrece la posibilidad de introducirnos plenamente en el intrincado mundo de la política, a través de este Objetivo la Moncloa. El juego consiste en elegir un partido (según sea éste variará la dificultad en el desarrollo), y conseguir alzarse con el mayor número de votos con el fin de mantener el gobierno de la nación en la siguiente legislatura, objetivo éste que se puede lograr mediante la buena administración de los bienes, la elección adecuada de los puntos fundamentales en que se ha de basar la política del partido, realizar unas campañas propagandísticas acertadas... en fin, cuidar de todos los aspectos sociales y políticos que hacen que un partido mantenga el mayor número de votos de los electores.

Además, durante el transcurso del juego tienen lugar una serie de acontecimientos inesperados que pueden cambiar, tanto positiva como negativamente, el normal desarrollo de una legislatura, como por ejemplo, donaciones al partido, o la huida del contable con gran parte de los fondos de la campaña.

Como veis, la idea general del juego resulta bastante atrayente, pero a la hora de ponerte delante del ordenador, dispuesto a pasar un rato entretenido, las cosas cambian ligéramente pues el juego mantiene una línea no demasiado brillante. La verdad es que es un poco aburrido contemplar la continua sucesión de tablas de porcentajes y listas de censos, sin apenas ningún gráfico que alegre la vista de vez en cuando.

A pesar de todo es un juego que recomendamos a la gente interesada en estos temas, pero avisamos de antemano que no se van a partir de risa precisamente.

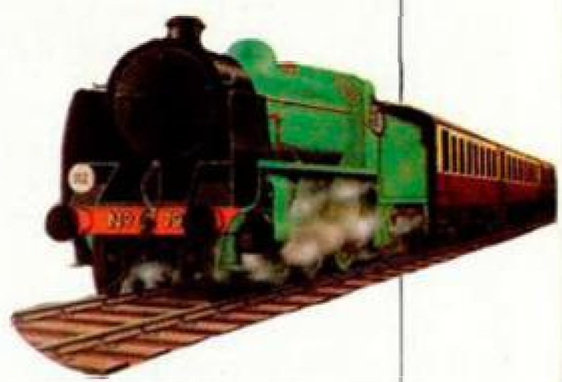




La vieja locomotora de vapor

SOUTHERN BELLE

Hewson Consultants
ERBE
Simulador



Las calderas se encuentran repletas de carbón, los depósitos de agua están a un nivel óptimo, todos los pasajeros están ya acomodados en sus asientos... La «Southern Belle» se dispone a efectuar su salida de la Estación Victoria con destino a Brighton. El largo recorrido a través de las verdes praderas de Inglaterra acaba de comenzar.

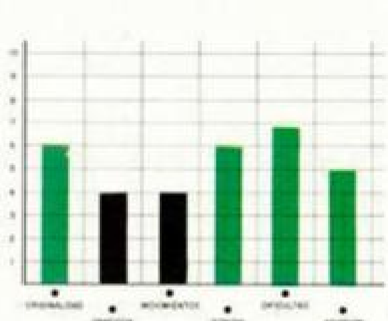


Lentamente, la locomotora se va deslizando sobre los desgastados raíles de la vieja estación, dejando tras de sí una leve cortina de vapor y un ambiente de tristeza y adioses.

Pero todas estas sensaciones de melancolía se disipan tan súbitamente como el vapor, cuando dejas de mirar hacia atrás y te das cuenta de que te encuentras ante una maraña de relojes, válvulas e indicadores que te recuerdan que de ti depende el que todo transcurra con normalidad y que no ocurra ningún percance durante el trayecto.

Para ello será necesario manejar una gran cantidad de controles: el indicador de agua, el de presión, el regulador de potencia, los frenos y todos los demás sistemas que hacen posible el funcionamiento de una auténtica máquina de vapor. Además de todo esto, los horarios establecidos deberán ser cumplidos con la mayor exactitud

Por otra parte, no todos los viajes son iguales, pues los inconvenientes que se pueden presentar provocan diferentes situaciones embarazosas que debemos resolver sobre la marcha. La ventaja de este juego —alguna tenía que tener— es que podremos elegir si queremos un viaje tranquilo o lo preferimos lleno de incidencias.



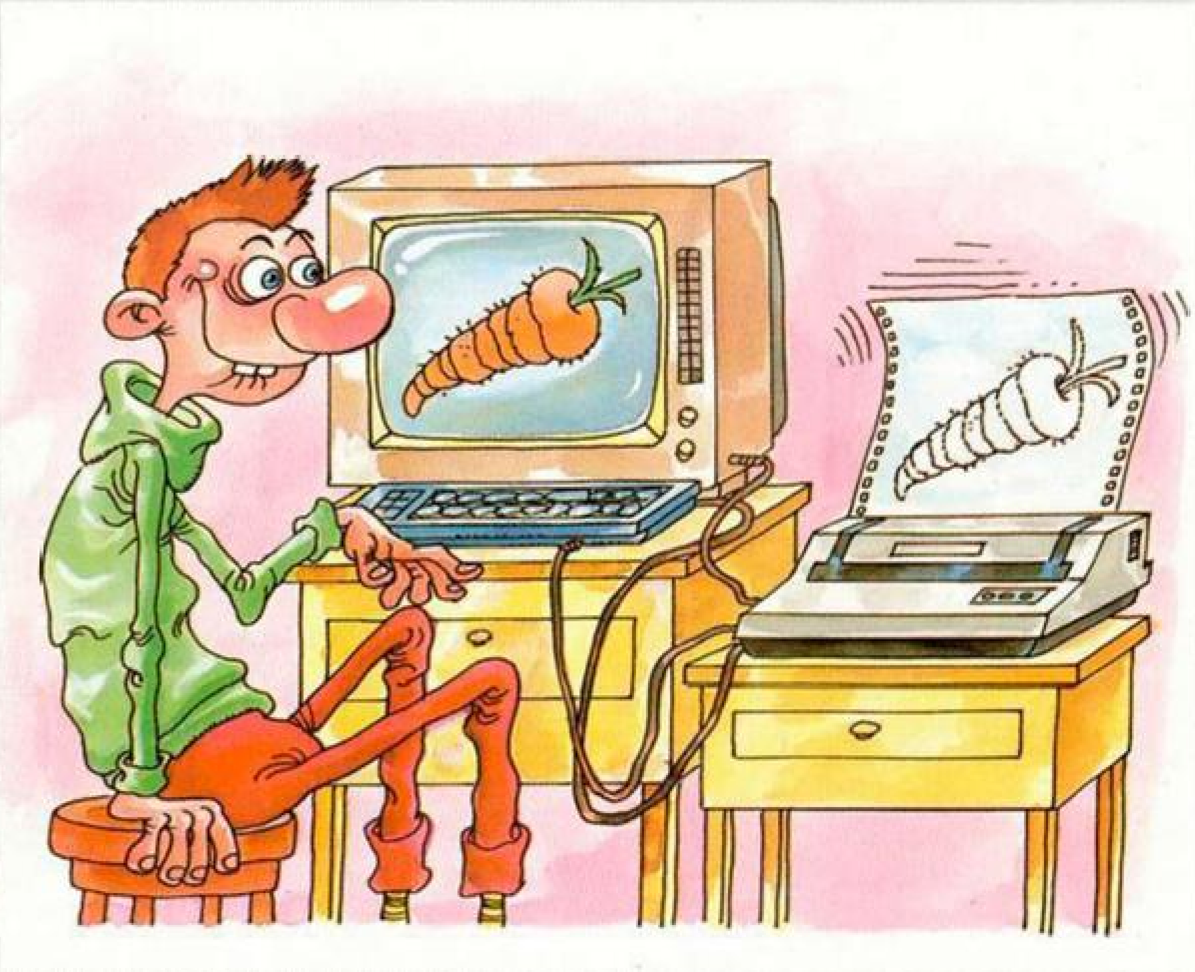
TRUCOS

COPIAS POR IMPRESORA

Miguel Moral nos manda un truco bastante útil a la hora de conseguir copias de toda la pantalla por la impresora.

En la dirección ØEAC (3756 en decimal) de la ROM se encuentra el comienzo de la rutina COPY. Esta rutina carga el registro B con 176 que es el númro de líneas en alta resolución de la parte superior de la pantalla, por lo que al ejecutar un COPY las dos líneas inferiores no son copiadas en la impresora.

Así pues, lo que propone Miguel es un corto programa en C/M que carga el registro B con 192, que son todas las líneas en alta resolución de la pantalla, e inmediatamente después se ejecuta un salto a la rutina COPY, pero después de la carga de B con 176.



A. PEREDA

```
DI;          / 243
LD B,CØ;/ 6,192
JP ØEAF;/ 195,175,14
```

En la izquierda se encuentra el programa en Ensamblador y a la derecha los POKES que hay que efectuar.

La rutina es relocalizable, además se notará su eficacia cuando exista algo en las dos líneas inferiores de la pantalla que se desee copiar. La llamada a la rutina debe hacerse inmediatamente después de la carga de la pantalla, por ejemplo:

```
Con una línea BASIC:
1Ø LOAD ""SCREEN$: RANDOMIZE USR dirección
    de la rutina
```

¿MEJOR EN BLANCO Y NEGRO?

Esta rutina que nos envía Iván Sansa en código máquina, convierte en blanco y negro cualquier pantalla tal como saldría en un «hardcopy» a impresora. De este modo podemos ver anticipadamente el resultado. Es completamente reubicable y sólo ocupa 16 bytes.

```
1Ø FOR x=Ø TO 15
2Ø READ n: POKE 65ØØØ+x,n: NEXT x
3Ø DATA 33,255,87,1,1,3,22,56,35,11,12Ø,177,
   2ØØ,114,24,248
4Ø RANDOMIZE USR 65ØØØ
```

CUESTION DE VELOCIDAD

Este truco de Jaime Munárriz puede ser de gran utilidad a la hora de intentar dotar de una mayor velocidad a un programa BASIC. Nos permite elegir las diferentes maneras de desempeñar una misma función de la manera más rápida, ya que nos proporciona con exactitud el tiempo que emplean en su funcionamiento.

```
1Ø POKE 23674,Ø: POKE 23673,Ø: POKE
   23672,Ø
2Ø FOR n=1 TO 1ØØØ
1ØØ Instruciones a comprobar
11Ø Instrucciones a comprobar
...
2ØØ NEXT n
21Ø PRINT (PEEK 23674 + 65536 + PEEK 23673 +
    256 + PEEK 23672)/50
```

Para conocer el tiempo real de unas líneas BASIC, habría que ejecutar primero el programa sin ninguna línea dentro del bucle, así conoceríamos el tiempo que éste tarda. Después incluiríamos las líneas que nos interesa probar y al tiempo obtenido le restaríamos el que tarda el bucle, y dividiríamos el resultado por mil.

De todas formas, para comparar las instrucciones no hace falta este cálculo, basta con ver el resultado que imprime la línea 210, y elegir el método que haya ofrecido un número menor.

En este espacio también tienen cabida los trucos que nuestros lectores quieran proponer.
Para ello, no tienen más que enviarlos por correo a MICROHOBBY, C/ La Granja, 8. Polígono Industrial de Alcobendas (Madrid).

ravillas desde el primer momento. Lo más importante es ir aprendiendo todo claramente; las "virguerías" podrá hacerlas luego cada uno, no obstante, a lo largo del curso tenemos reservadas para nuestros lectores, maravillosas sorpresas.

Vamos con nuestro siguiente ejemplo, esta vez vamos a leer desde código máquina un número que habremos almacenado desde Basic en la variable del Sistema "SEED". Se trata de leer el contenido de SEED y sacarlo a pantalla a través del registro "BC". En esta ocasión, almacenaremos el programa a partir de la dirección de memoria 30000, para lo cual, bajaremos primero la RAMTOP a 29999. Estas direcciones son válidas tanto para los usuarios de 16 K como de 48 K.

Nuestro programa es el siguiente:

```
      ORG 30000
      LD HL,(SEED)
      LD B,H
      LD C,L
      RET
SEED  EQU #5C76
```

La primera línea: "ORG 300 00" es un pseudo-nemónico, no se puede ensamblar y su única finalidad es indicarle al ensamblador que deberá ensamblar el programa a partir de la dirección 30000.

La última línea "SEED EQU #5C76" tampoco se puede ensamblar, se trata de una definición de etiqueta, su finalidad es asignarle a la etiqueta "SEED" el valor 5C76h (23670). El programa simplificado, quedaría:

```
LD HL,(#5C76)
LD B,H
LD C,L
RET
```

Para codificarlo, tomamos de nuevo las tablas y buscamos el código de:

```
LD HL,(nn)
```

que resulta ser 2Ah (42). A continuación, vendrá el operando invertido: 76h (118) y 5Ch (92). Ahora buscamos:

```
LD B,H
```

y

```
LD C,L
```

cuyos códigos resultan ser respectivamente: 44h (68) y 4Dh (77). Finalmente, ponemos el código de RET, es decir, C9h (201). Nuestro programa queda de la siguiente forma:

```
2A,76,5C,44,4D,C9
```

O escrito en decimal:

```
42,118,92,68,77,201
```

Vamos a construir el programa Basic que lo introduce en memoria y lo ejecuta:

```
10 CLEAR 29999
20 FOR n=30000 TO 30005
30 READ a: POKE n,a: NEXT n
40 DATA 42,118,92,68,77,201
50 INPUT "SEED";a
60 RANDOMIZE a
70 PRINT USR 30000
```

La línea 10 baja la RAMTOP para preservar nuestro programa contra borrados accidentales. Las líneas 20 y 30 cargan en memoria el programa que se encuentra en los datos de la línea 40. La línea 50 nos pide un valor para SEED, y la línea 60 lo introduce en la variable "SEED" siempre que este valor no sea cero. Finalmente, la línea 70 ejecuta nuestro programa en código máquina e imprime en pantalla el resultado.

En este ejemplo, vemos que es posible establecer una comunicación bidireccional entre Basic y Código Máquina para transferir datos; existen otras muchas formas de realizar esta comunicación que se irán viendo en ejemplos sucesivos.

En nuestro tercer ejemplo, vamos a leer la variable del Sistema RAMTOP desde código máquina, y utilizaremos la pila para sacarla a pantalla por el registro "BC". Asimismo, veremos cómo almacenar una rutina en código máquina dentro de una línea REM del programa Basic.

Primero leeremos el contenido de la variable RAMTOP, cargándolo sobre el registro "HL", luego trasferiremos este contenido al "BC" a través de la pila; el programa podría ser el siguiente:

```
        LD HL,(RAMTOP)
        PUSH HL
        POP BC
        RET
RAMTOP  EQU 23730
```

De nuevo, utilizamos una etiqueta que definimos en la última línea, antes de codificar el programa, eliminamos la etiqueta, quedando el programa simplificado:

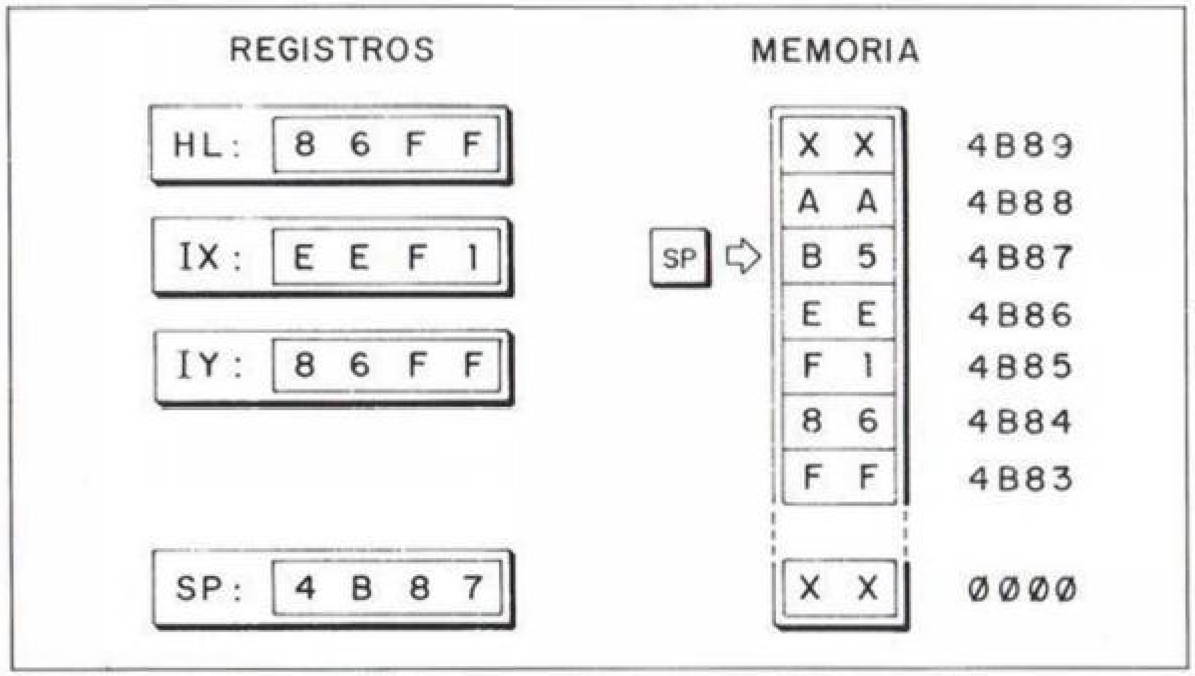


Fig. 5-1F. Contenidos, después de ejecutar la instrucción: «POP IX».

ha subido dos casillas, para apuntar al dato anteriormente introducido.

Una observación interesante, es que el contenido de las casillas que componen la pila, no se ha modificado; el número 86FFh sigue estando ahí, aunque a nosotros nos da igual, recuérdese que sólo podemos acceder cada vez, al dato señalado por el puntero; Las casillas 4B84h y 4B83h que contienen el dato 86FFh, sólo se borrarán totalmente cuando la pila vuelva a expandirse.

Ahora, vamos a recuperar el siguiente dato de la pila y lo asignaremos al registro "IX":

```
POP IX
```

Esta instrucción toma el último dato de la pila y lo asigna al registro "IX"; recuérdese que el último dato, es siempre el que es apuntado por el puntero. Recuérdese también, que la dirección de la casilla apuntada por el puntero es el contenido del registro "SP"; las letras "S" y "P", son las iniciales de "Stack Pointer", en Inglés, "Puntero de Pila".

La FIGURA 5-1F, muestra los contenidos después de ejecutar la instrucción "POP IX", vemos otra vez que los datos de la pila no se han perdido, pero para nosotros no existen, ya que el puntero ha vuelto a subir dos casillas y ahora se considera que el último dato de la pila es AAB5h.

Vamos a recuperar, por último, este dato:

```
POP IY
```

Esta instrucción, toma el último dato de la pila y lo asigna al registro "IY".

En la FIGURA 5-1G, vemos la situación final, el dato AAB5h ha sido asignado al registro "IY" y el puntero se ha vuelto a incrementar dos veces, para apuntar al mismo sitio que lo hacía al principio de estos ejemplos.

Ya hemos sacado de la pila todos los datos que habíamos introducido. El puntero ha quedado en la misma posición donde estaba al principio. Si nuestro ejemplo fuese una subrutina de un programa, en la dirección donde apunta ahora el puntero se encontraría almacenada la dirección de retorno y ahora sería posible retornar al punto desde donde se llamó a esta subrutina.

Con la pila se pueden hacer muchas cosas. Supongamos que en Basic queremos intercambiar el contenido de dos variables "a" y "b"; en ese caso, necesitamos generar una tercera variable que nos sirve de "puente", de la forma:

```
10 LET c=a
20 LET a=b
30 LET b=c
```

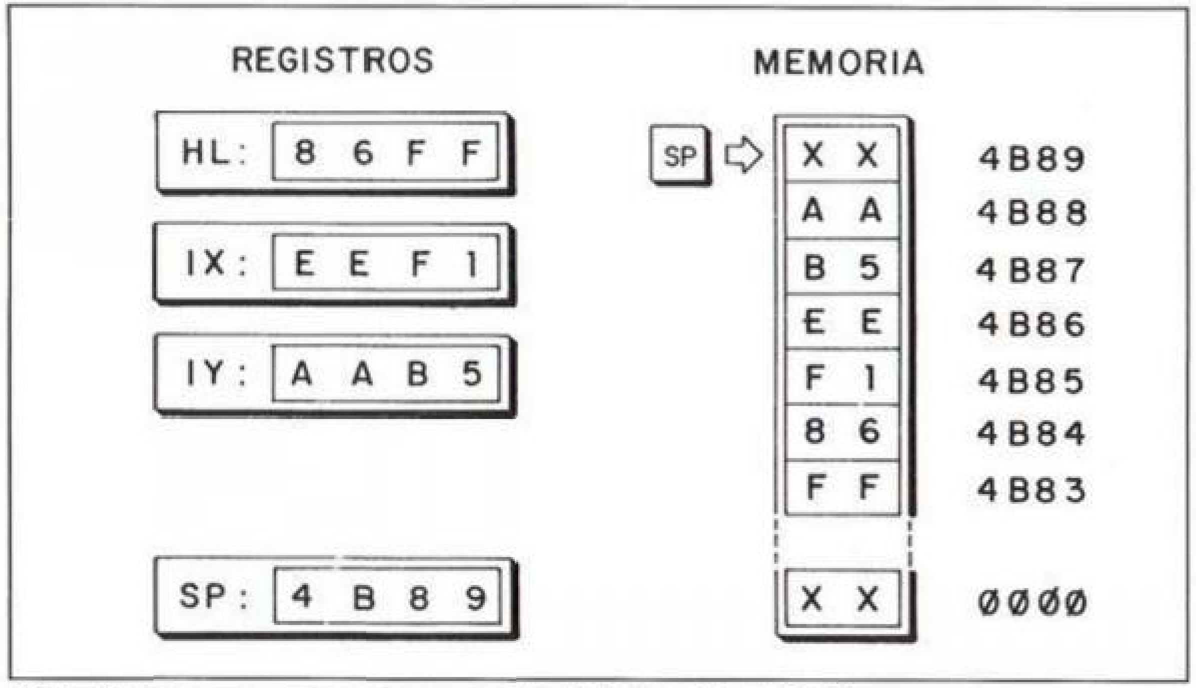


Fig. 5-1G. Contenidos, después de ejecutar la instrucción: «POP IY».

Trabajando en código máquina, podemos intercambiar el contenido de dos registros, utilizando la pila en lugar de la variable "puente" del Basic, supongamos que queremos intercambiar los contenidos de los registros "BC" y "DE"; podríamos hacer:

```
PUSH BC
PUSH DE
POP BC
POP DE
```

Por supuesto, esto es sólo un ejemplo, no se quedan aquí las utilidades de la pila. Su función principal es la de salvar temporalmente, el contenido de algún registro, mientras se utiliza para algo y luego, restituirle, de nuevo su contenido. No obstante, con la pila se pueden hacer muchas más cosas, en capítulos posteriores veremos cómo utiliza la pila el Intérprete de Basic, para poder retornar desde cualquier punto en caso de error.

Tablas de codificación:

A continuación, vamos a ver una serie de tablas que nos han de servir para codificar las instrucciones rápidamente, cuando ensamblemos a mano.

En las tablas se ha representado el código máquina de cada instrucción, tanto en decimal como en binario.

Cuando el código máquina ocupa más de un byte, se han puesto uno a continuación del otro, separados por comas.

Donde pone "d", se entiende que ese byte va ocupado por un entero de desplazamiento en complemento a dos.

Donde pone "n", debe ir el operando "n" que aparece en el código fuente de la instrucción.

Donde aparecen dos bytes seguidos con "n", debe ir el operando "nn" del código fuente de la instrucción; primero irá el octeto menos significativo y luego el más significativo; por ejemplo: supongamos que el operando "nn" fuera 2A4Bh, primero iría 4Bh y luego 2Ah.

En el apartado de ejemplos, veremos con claridad la forma de ensamblar a mano pequeños programas.

La disposición de las tablas es la siguiente:

curso. Hasta ese momento, suponemos que los programas se ejecutan en un orden puramente secuencial, desde la primera instrucción hasta la última.

Ejemplos

A continuación, vamos a ver una serie de ejemplos prácticos que el lector podrá introducir en su ordenador, tanto si dispone de ensamblador, como si no.

A través de estos ejemplos, se pretende no sólo aprender a utilizar las instrucciones de carga, sino también, aprender a ensamblar un programa "a mano" y cargarlo desde Basic en cualquier lugar de la memoria.

Antes de eso, y como nota previa, vamos a ver la forma de retornar a Basic desde código máquina cuando finalice la ejecución de cada uno de nuestros programas. En general, llamaremos a nuestros programas con la función USR del Basic, esta función ejecutará nuestras rutinas como si se tratase de subrutinas del sistema operativo, por lo que el procedimiento de retornar a éste, será el mismo que para retornar desde cualquier subrutina, es decir, la instrucción "RET" que se ensambla como C9h (2Ø1) y es equivalente al RETURN del Basic.

Quizá esto se comprenda mejor cuando estudiemos las subrutinas en código máquina. Por ahora, nos basta con saber que al final de cada uno de nuestros programas, deberá ir la instrucción RET.

Empecemos por lo más sencillo, vamos simplemente a cargar un número en el registro "BC". Escogemos este registro, por que es su contenido el que nos devuelve la función USR cuando retornamos a Basic.

Nuestro primer programa en código máquina podría ser el siguiente:

```
LD BC,27263
RET
```

Que también podría haberse escrito en hexadecimal de la siguiente forma:

```
LD BC,#6A7F
RET
```

Vamos a ensamblar a mano este sencillo ejemplo, y luego lo cargaremos en uno de los lugares que indicábamos en el capítulo 4, el buffer de impresora.

Cogemos las tablas de codificación, y vemos que la instrucción

```
LD BC,nn
```

tiene el código Ø1(1), de forma que éste será el primer byte de nuestro programa.

A continuación, deberemos poner el operando de dos bytes "nn", con el orden de los octetos invertido. Como en este caso, el operando es 6A7F, deberemos poner primero 7Fh (127) y luego, 6Ah (1Ø6). Finalmente, pondremos el código de RET, C9h (2Ø1).

Nuestro programa queda, por tanto, de la siguiente forma:

```
Ø1,7F,6A,C9
```

O escrito en decimal:

```
1,127,1Ø6,2Ø1
```

Hasta ahora, hemos hecho todo esto sobre el papel; por fin llega el momento de poner en marcha nuestro querido Spectrum.

Para introducir los cuatro valores que componen nuestro código máquina, podemos POKEarlos en memoria ayudándonos de un bucle FOR... NEXT:

```
10 FOR n=23296 TO 23299
20 READ a: POKE n,a
30 NEXT n
40 DATA 1,127,106,201
50 PRINT USR 23296
```

Nuestro programa está en los datos de la línea 4Ø, las líneas 1Ø, 2Ø y 3Ø los van introduciendo secuencialmente en memoria, finalmente la línea 5Ø lo ejecuta imprimiendo el resultado de USR en el retorno.

Teclee el programa, revise que no haya habido errores, y pulse RUN...

Si todo ha ido correctamente, deberá ver el número 27263 en la esquina superior izquierda de la pantalla.

No parece un resultado muy espectacular comparado con los prodigios semimágicos que se suelen esperar del código máquina. Ciertamente, no se puede pretender más con cuatro bytes, un simple "PRINT a" de Basic implica la ejecución de cientos de instrucciones en código máquina.

No debe desanimarse el lector ni pretender hacer ma-

Código Fuente	Hexadecimal	Decimal
LD SP,HL	F9	249
LD SP,IX	DD,F9	221,249
LD SP,IY	FD,F9	253,249

Fig. 5-8. Grupo de carga de registro «SP».

Código Fuente	Hexadecimal	Decimal
PUSH BC	C5	197
PUSH DE	D5	213
PUSH HL	E5	229
PUSH AF	F5	245
PUSH IX	DD,E5	221,229
PUSH IY	FD,E5	253,229
POP BC	C1	193
POP DE	D1	209
POP HL	E1	225
POP AF	F1	241
POP IX	DD,E1	221,225
POP IY	FD,E1	253,225

Fig. 5-9. Grupo de manejo de pila.

Carga del registro "PC"

Seguramente, el lector ya se habrá dado cuenta de que no hemos mencionado en ninguna instrucción al registro "PC"; este hecho se debe a que se trata de un registro especial, que tiene asignada una función muy específica.

El registro "PC" o "Contador de Programa", contiene siempre la dirección en memoria de la siguiente instrucción a ejecutar, por lo que el hecho de cargarlo con un número, implica que la siguiente instrucción será leída desde la posición de memoria apuntada por ese número, es decir, se producirá un salto o bifurcación en el flujo del programa. Vamos a verlo con un ejemplo.

Supongamos que acabamos de leer una instrucción de tres bytes de longitud, que ocupaba las posiciones 40000, 40001 y 40002. En este momento, el registro "PC" contiene el valor 40003 que es la dirección desde donde se leerá la siguiente instrucción; si la instrucción que estamos ejecutando, modifica el contenido del "PC", digamos que lo pone a 60000, la siguiente instrucción será leída desde esta dirección, con lo que se habrá producido un salto en el programa.

Los saltos y bifurcaciones tienen una importancia tan grande en cualquier lenguaje, que se ha reservado un grupo de instrucciones para este fin; se trata del grupo de instrucciones de "cambio de secuencia", que se estudiarán en el capítulo 10 de este

Código Fuente	Hexadecimal	Decimal	Código Fuente	Hexadecimal	Decimal
LD A,A	7F	127	LD H,E	63	99
LD A,B	78	120	LD H,H	64	100
LD A,C	79	121	LD H,L	65	101
LD A,D	7A	122	LD L,A	6F	111
LD A,E	7B	123	LD L,B	68	104
LD A,H	7C	124	LD L,C	69	105
LD A,L	7D	125	LD L,D	6A	106
LD B,A	47	71	LD L,E	6B	107
LD B,B	40	64	LD L,H	6C	108
LD B,C	41	65	LD L,L	6D	109
LD B,D	42	66	LD A,n	3E,n	62,n
LD B,E	43	67	LD B,n	06,n	6,n
LD B,H	44	68	LD C,n	0E,n	14,n
LD B,L	45	69	LD D,n	16,n	22,n
LD C,A	4F	79	LD E,n	1E,n	30,n
LD C,B	48	72	LD H,n	26,n	38,n
LD C,C	49	73	LD L,n	2E,n	46,n
LD C,D	4A	74	LD A,(HL)	7E	126
LD C,E	4B	75	LD B,(HL)	46	70
LD C,H	4C	76	LD C,(HL)	4E	78
LD C,L	4D	77	LD D,(HL)	56	86
LD D,A	57	87	LD E,(HL)	5E	94
LD D,B	50	80	LD H,(HL)	66	102
LD D,C	51	81	LD L,(HL)	6E	110
LD D,D	52	82	LD A,(IX+d)	DD,7E,d	221,126,d
LD D,E	53	83	LD B,(IX+d)	DD,46,d	221,70,d
LD D,H	54	84	LD C,(IX+d)	DD,4E,d	221,78,d
LD D,L	55	85	LD D,(IX+d)	DD,56,d	221,86,d
LD E,A	5F	95	LD E,(IX+d)	DD,5E,d	221,94,d
LD E,B	58	88	LD H,(IX+d)	DD,66,d	221,102,d
LD E,C	59	89	LD L,(IX+d)	DD,6E,d	221,110,d
LD E,D	5A	90	LD A,(IY+d)	FD,7E,d	253,126,d
LD E,E	5B	91	LD B,(IY+d)	FD,46,d	253,70,d
LD E,H	5C	92	LD C,(IY+d)	FD,4E,d	253,78,d
LD E,L	5D	93	LD D,(IY+d)	FD,56,d	253,86,d
LD H,A	67	103	LD E,(IY+d)	FD,5E,d	253,94,d
LD H,B	60	96	LD H,(IY+d)	FD,66,d	253,102,d
LD H,C	61	97	LD L,(IY+d)	FD,6E,d	253,110,d
LD H,D	62	98			

Fig. 5-2. Grupo de carga en registros.

Código Fuente	Hexadecimal	Decimal
LD (HL),A	77	119
LD (HL),B	7Ø	112
LD (HL),C	71	113
LD (HL),D	72	114
LD (HL),E	73	115
LD (HL),H	74	116
LD (HL),L	75	117
LD (IX+d),A	DD,77,d	221,119,d
LD (IX+d),B	DD,7Ø,d	221,112,d
LD (IX+d),C	DD,71,d	221,113,d
LD (IX+d),D	DD,72,d	221,114,d
LD (IX+d),E	DD,73,d	221,115,d
LD (IX+d),H	DD,74,d	221,116,d
LD (IX+d),L	DD,75,d	221,117,d
LD (IY+d),A	FD,77,d	253,119,d
LD (IY+d),B	FD,7Ø,d	253,112,d
LD (IY+d),C	FD,71,d	253,113,d
LD (IY+d),D	FD,72,d	253,114,d
LD (IY+d),E	FD,73,d	253,115,d
LD (IY+d),H	FD,74,d	253,116,d
LD (IY+d),L	FD,75,d	253,117,d
LD (HL),n	36,n	54,n
LD (IX+d),n	DD,36,d,n	221,54,d,n
LD (IY+d),n	FD,36,d,n	253,54,d,n

Fig. 5-3. Grupo de carga de memoria.

Código Fuente	Hexadecimal	Decimal
LD A,(BC)	ØA	1Ø
LD A,(DE)	1A	26
LD A,(nn)	3A,n,n	58,n,n
LD A,I	ED,57	237,87
LD A,R	ED,5F	237,95

Fig. 5-4. Grupo de carga en acumulador.

Código Fuente	Hexadecimal	Decimal
LD (BC),A	Ø2	2
LD (DE),A	12	18
LD (nn),A	32,n,n	5Ø,n,n
LD I,A	ED,47	237,71
LD R,A	ED,4F	237,79

Fig. 5-5. Grupo de salvar acumulador.

Código Fuente	Hexadecimal	Decimal
LD BC,nn	Ø1,n,n	1,n,n
LD DE,nn	11,n,n	17,n,n
LD HL,nn	21,n,n	33,n,n
LD SP,nn	31,n,n	49,n,n
LD IX,nn	DD,21,n,n	221,33,n,n
LD IY,nn	FD,21,n,n	253,33,n,n
LD HL,(nn)	2A,n,n	42,n,n
LD BC,(nn)	ED,4B,n,n	237,75,n,n
LD DE,(nn)	ED,5B,n,n	237,91,n,n
LD SP,(nn)	ED,7B,n,n	237,123,n,n
LD IX,(nn)	DD,2A,n,n	221,42,n,n
LD IY,(nn)	FD,2A,n,n	253,42,n,n

Fig. 5-6. Grupo de carga en registros de 16 bits.

Código Fuente	Hexadecimal	Decimal
LD (nn),HL	22,n,n	34,n,n
LD (nn),BC	ED,43,n,n	237,67,n,n
LD (nn),DE	ED,53,n,n	237,83,n,n
LD (nn),SP	ED,73,n,n	237,115,n,n
LD (nn),IX	DD,22,n,n	221,34,n,n
LD (nn),IY	FD,22,n,n	253,34,n,n

Fig. 5-7. Grupo de carga en memoria de 16 bits.

PARA PERSONAS CON VISION DE FUTURO

ELECTRONICA

INSTALADOR ELECTRICISTA GENERAL

VIDEO NUEVO

FONTANERIA

MAESTRO ALBAÑIL

DELINEANTE GENERAL

GRADUADO ESCOLAR

BASICO DE PSICOLOGIA

PINTURA AL OLEO

DIBUJO Y PINTURA

CONTABILIDAD NUEVO

INTRODUCCION A LA INFORMATICA

CORTE Y CONFECCION NUEVO

PUERICULTURA

DECORACION

FOTOGRAFIA

MECANICO DE AUTOMOVILES

MECANICO DE MOTOS NUEVO

Nuestra sociedad resulta cada día más competitiva, la mayor tecnificación de los trabajos, la falta de empleo, etc., hacen que se nos exija un mayor nivel de conocimientos.

Las personas con "vista" ocupan sus ratos de ocio en obtener conocimientos que les resulten productivos o gratificantes. De esta forma y con una clara **visión del futuro** van logrando abrirse camino en la vida.

En CEAC queremos ayudarle a que usted obtenga el máximo partido de su tiempo; que logre la mejora profesional deseada o las mayores satisfacciones de su hobby. Para esto ponemos a su disposición una moderna y completa gama de Cursos y la ayuda de expertos profesionales que le atenderán durante todo su Curso. **¡Decídase a ver más claro el futuro!**

NUEVO

CURSO DE BASIC + MICROORDENADORES



Un Curso ideal para personas que deseen formarse rápida y eficazmente en la PROGRAMACION DE ORDENADORES en lenguaje BASIC. Aprenderá practicando desde la primera lección, en un ordenador personal que, caso de no tenerlo ya, CEAC proporciona junto al Curso. Y todo ello sin moverse de su domicilio. Todo lo que necesita es el **ordenador** y el Curso **BASIC + MICROORDENADORES.** Infórmese. **SEA USTED UNO DE LOS PRIMEROS.**

CEAC GARANTIZA El pleno aprovechamiento de sus estudios o le devolvemos SU DINERO

SOLICITE INFORMACION ENVIANDO EL CUPON ADJUNTO O DIRIJA SUS CARTAS A:

...o llame al teléfono (93) 245 33 06 de Barcelona

CEAC

CENTRO DE ENSEÑANZA A DISTANCIA AUTORIZADO POR EL MINISTERIO DE EDUCACION Y CIENCIA N.º 8039185 (BOLETIN OFICIAL DEL ESTADO 3-6-83) Aragón, 472 - 08013 Barcelona

843

RELLENE Y ENVIE ESTE CUPON HOY MISMO

CURSOS CEAC

- ELECTRONICA (con experimentos)
- GRADUADO ESCOLAR
- CORTE Y CONFECCION
- FOTOGRAFIA
- DECORACION
- BASICO DE PSICOLOGIA
- INTRODUCCION A LA INFORMATICA
- JARDINERIA
- MECANICO DE MOTOS
- DELINEANTE GENERAL
- VIDEO
- BASIC + MICROORDENADORES

PUERICULTURA
- Puericultura
- Educación Preescolar
- Puericultura y Educación Preescolar

CONTABILIDAD
- Contabilidad
- Jefe de Contabilidad
- Contabilidad y Control Presupuestario

CONSTRUCCION
- Técnico en Construcción
- Maestro Albañil
- Fontanería
- Delineante Construcción

MOTOR Y AUTOMOVIL
- Mecánico de Automóviles
- Jefe Taller de Automóviles
- Electricidad del Automóvil

DIBUJO Y PINTURA
- Pintura al Oleo
- Dibujo Artístico
- Dibujo y Pintura
- Dibujante General

ELECTRICIDAD
- Instalador Electricista General
- Maestro Electricista
- Fontanería y Electricidad

GRATUITAMENTE

Sr. Director: Deseo recibir detallada información del Curso de ____________

Nombre y apellidos ____________ Edad ______

Domicilio ____________

Bloque ______ N.º ____ Piso ____ Puerta ______ Teléfono ______

Código Postal ______ Población ______

Provincia ______ Profesión ______

CEAC Aragón, 472 (Dpto. X – SA) 08013 Barcelona

VARIADOR DE VELOCIDAD

José M. LAZO

Esta es una subrutina corta y versátil que permite reducir la velocidad de algunos programas comerciales y de la gran mayoría de los que se pueden desarrollar en Basic.

La subrutina se ubica en los útimos 512 octetos de la memoria por lo tanto puede ser utilizada con cualquier programa que no use estas posiciones para su funcionamiento, tanto en código máquina como en Basic, siempre y cuando no deshabilite las interrupciones ya que el variador de velocidad se basa en una interrupción.

Manejo de la rutina

Su manejo es muy sencillo. Después de haberla cargado en la memoria desde una cinta (naturalmente, primero habrá que teclearla de la revista), aparecerá en pantalla un texto explicativo con dos cifras que indican las direcciones de activación y desactivación de la rutina.

RANDOMIZE USR 65ØØØ - ACTIVA LA RUTINA

RANDOMIZE USR 65ØØ7 - DESACTIVA LA RUTINA

Al presionar cualquier tecla se producirá un NEW. No hay que alarmarse, ya que la rutina está por encima de la RAM TOP. Para ponerla en funcionamiento haz RANDOMIZE USR 65ØØØ. Pulsando ahora el 1 y el 2 simultáneamente, obtendremos una primera reducción de la velocidad.

Podemos observar que a partir de ahora el borde de nuestra pantalla tiene dos colores: uno rojo en la parte superior y otro en la parte inferior que coincide con el que tuviéramos antes. No hay que extranãrse por ello ya que es el indicador de la velocidad seleccionada. La parte superior roja se hará más grande cuanto mayor lentitud imprimamos al programa. Pero pasemos a explicar como se maneja...

Dijimos anteriormente que pulsando el 1 y el 2 juntos entrábamos en la interrupción. Hazlo si no lo has hecho todavía.

Para probrar el efecto vamos a elaborar un sencillo ejemplo en Basic. Teclea el siguiente programa con la interrupción ya accionada, esto es, con la parte superior del borde en rojo.

FOR N=1 TO 7Ø4: PRINT "Ø";: NEXT N

Ejecútalo con RUN y la pantalla se llenará de ceros.

Repite la operación, pero esta vez, cuando sólo esté completa la mitad de la pantalla, pulsa simultáneamente las teclas Symbol Shift y Enter: el programa en Basic se ha parado. Si sueltas las dos teclas se volverá a poner en marcha.

Ahora haz lo mismo, pulsando Enter y Symbol Shift, pero a la vez pulsa el 2, podrás observar que el borde rojo va creciendo hasta llegar a la parte de abajo y que el bucle en Basic se ralentiza: de esta forma disminuimos la velocidad. Pulsando el 1 en vez del 2, obtendremos el efecto contrario.

Si quieres ahora puedes introducir en el ordenador otro programa cualquiera; si éste cumple las condiciones arriba indicadas el variador funcionará.

Pero hay otros comandos útiles, a saber:

Pulsando el 3 de la misma manera no se modifica la velocidad que tengamos fijada pero sí, en cambio, desaparece el efecto de borde. Esto sirve para los programas comerciales que utilizan el borde, evitando un molesto parpadeo.

Pulsando el 1 y el 2 recuperamos de nuevo el efecto inicial.

Pulsando el 9 se eliminan todas las pausas, disponiendo de la máxima velocidad. Con el 1 y el 2 volvemos otra vez al efecto de borde y la velocidad que tengamos prefijada.





LISTADO ASSEMBLER DEL VARIADOR DE VELOCIDAD

```
10 ;Programa regulador de
20 ;velocidad
30 ;J.M.Lazo
40         ORG  65000
50 INA     LD   A,253
60         LD   I,A
70         IM   2
80         RET
90 DESA    IM   1
100        RET
110        ORG  65023
120        DEFW COMIEN
130 COMIEN PUSH AF
140        PUSH BC
150        PUSH DE
160        PUSH HL
170        JP   ESTADO
180 START  LD   A,127
190        IN   A,(#FE)
200        BIT  1,A
210        JR   NZ,BORDE
220        LD   A,191
230        IN   A,(#FE)
240        BIT  0,A
250        JR   NZ,BORDE
260        LD   A,247
270        IN   A,(#FE)
280        BIT  0,A
290        JR   Z,DECRE
300        BIT  1,A
310        JR   Z,INCRE
320        BIT  2,A
330        JP   Z,NOBORD
340        LD   A,239
350        IN   A,(#FE)
360        BIT  0,A
370        JP   Z,NEW
380        BIT  1,A
390        JP   Z,QUIT
400        JR   START
410 BORDE  CALL DIBUJO
420        LD   A,2
430        OUT  (#FE),A
440        LD   HL,(WAIT)
450 LOOP5  DEC  HL
460        LD   A,H
470        OR   L
480        JR   NZ,LOOP5
490        LD   A,(23624)
500        SRA  A
510        SRA  A
520        SRA  A
530        SET  3,A
540        SET  4,A
550        RES  3,A
560        RES  4,A
570        OUT  (#FE),A
580        LD   A,(PAUSE)
590        SRA  A
600        LD   B,A
610 LOOP1  CALL PAUSA
620        DJNZ LOOP1
630 OUT    POP  HL
640        POP  DE
650        POP  BC
660        POP  AF
670        RST  #38
680        RETI
690 DECRE  LD   A,(PAUSE)
700        DEC  A
710        CP   1
720        JR   NZ,DOS
730        INC  A
740 DOS    LD   (PAUSE),A
750        JR   BORDE
760 INCRE  LD   A,(PAUSE)
770        INC  A
780        CP   111
790        JR   NZ,DOS
800        DEC  A
810        JR   DOS
820 PAUSE  DEFB 2
830 PAUSA  PUSH BC
840        LD   B,8
850 LOOP3  DJNZ LOOP3
860        POP  BC
870        RET
880 ESTADO LD   A,(ESTBOR)
890        CP   0
900        JR   Z,BORDEN
910        LD   A,(COSA)
920        CP   0
930        JP   NZ,START
940        LD   A,247
950        IN   A,(#FE)
960        BIT  0,A
970        JR   NZ,OUT
980        BIT  1,A
990        JR   NZ,OUT
1000       LD   A,1
1010       LD   (COSA),A
1020       JP   START
1030 QUIT  LD   A,0
1040       LD   (COSA),A
1050       JR   OUT
1060 COSA  DEFB 0
1070 DIBUJO LD  B,13
1080       SCF
1090       CCF
1100       LD   A,(PAUSE)
1110       LD   H,0
1120       LD   L,A
1130       PUSH HL
1140       POP  DE
1150 LOOP7 ADC  HL,DE
1160       DJNZ LOOP7
1170       LD   DE,492
1180       ADD  HL,DE
1190       LD   (WAIT),HL
1200       RET
1210 WAIT  DEFW 500
1220 NOBORD LD  A,0
1230       LD   (ESTBOR),A
1240       JR   OUT
1250 BORDEN LD  A,(PAUSE)
1260       LD   B,A
1270 LOOP11 PUSH BC
1280       LD   B,41
1290 LOOP10 DJNZ LOOP10
1300       POP  BC
1310       DJNZ LOOP11
1320       LD   A,247
1330       IN   A,(#FE)
1340       BIT  0,A
1350       JP   NZ,OUT
1360       BIT  1,A
1370       JP   NZ,OUT
1380       LD   A,1
1390       LD   (ESTBOR),A
1400       JP   OUT
1410 ESTBOR DEFB 1
1420 NEW   LD   HL,23730
1430       LD   (HL),239
1440       INC  HL
1450       LD   (HL),250
1460       INC  HL
1470       LD   (HL),255
1480       INC  HL
1490       LD   (HL),255
1500       JP   #11B7
1510 ZINAL ENT  INA
```

Pulsando el Ø siempre del mismo modo (con Symbol Shift + Enter) hacemos un New de la memoria, pero la subrutina no se pierde, la podremos disparar otra vez con RANDOMIZE USR 65ØØØ.

Hasta aquí el manejo, ahora vamos a ver el funcionamiento.

Funcionamiento del programa

Con ayuda del listado assembler podemos seguir el flujo del programa y

comprender claramente su funcionamiento.

5Ø INA. Se encarga de activar la interrupción.

9Ø DESA. Desactiva la interrupción volviendo a la normalidad.

13Ø COMIEN. Primero guarda los registros que vamos a utilizar en la pila para no corromper sus valores; antes de volver de la interrupción hay que recuperarlos en orden inverso a como lo hemos metido.

18Ø START. Inicia un muestreo del teclado saltando a distintos bloques de la interrupción según las teclas que estén pulsadas. Si están accionadas Simbol Shift y Enter, entra en un bucle cerrado hasta que las soltemos.

41Ø BORDE. Coloca un trozo de borde de color rojo proporcionalmente al valor que tenga la variable PAUSE. Esta subrutina sólo puede ser llamada una vez dentro de la interrupción para que así coincida con el barrido que hace la ULA en la pantalla del monitor. Llama a la rutina DIBUJO que calcula el valor que tendrá WAIT. Luego hace un OUT de valor 2 que pone el borde rojo, esperando en un bucle según esté WAIT y pone el borde del color normal.

63Ø OUT. Aquí salimos de la interrupción. Saca de la pantalla los valores de los registros, hace una llamada a una subrutina de la ROM que lee el teclado y volvemos con la instrucción RETI.

69Ø DECRE. Esta subrutina es llamada desde el muestreo del teclado para decrementar el valor de la variable PAUSE.

74Ø DOS. Almacena el nuevo valor de PAUSE y llama a BORDE.

76Ø INCRE. Aquí hacemos lo contrario que en DECRE.

83Ø PAUSA. Hace una pausa. Esta subrutina es llamada sólo cuando tenemos activado el efecto de borde, desde la línea 61Ø.

88Ø ESTADO. Es llamada desde el principio de la interrupción. Bifurca a distintas partes del programa según estemos utilizando el borde o no; aquí también hay un muestreo del teclado.

1Ø3Ø QUIT. Pone la interrupción en el estado de sin pausa ni efecto de borde.

1Ø7Ø DIBUJO. Esta subrutina es el corazón central para conseguir un efecto de borde acorde con el valor de la pausa que tengamos seleccionado. Coge el valor de PAUSE que puede ir de 1 a 255, lo multiplica por 13, le suma 492 y actualiza la variable WAIT con este valor.

122Ø NOBORD. Pone la interrupción en el estado con pausa y sin borde.

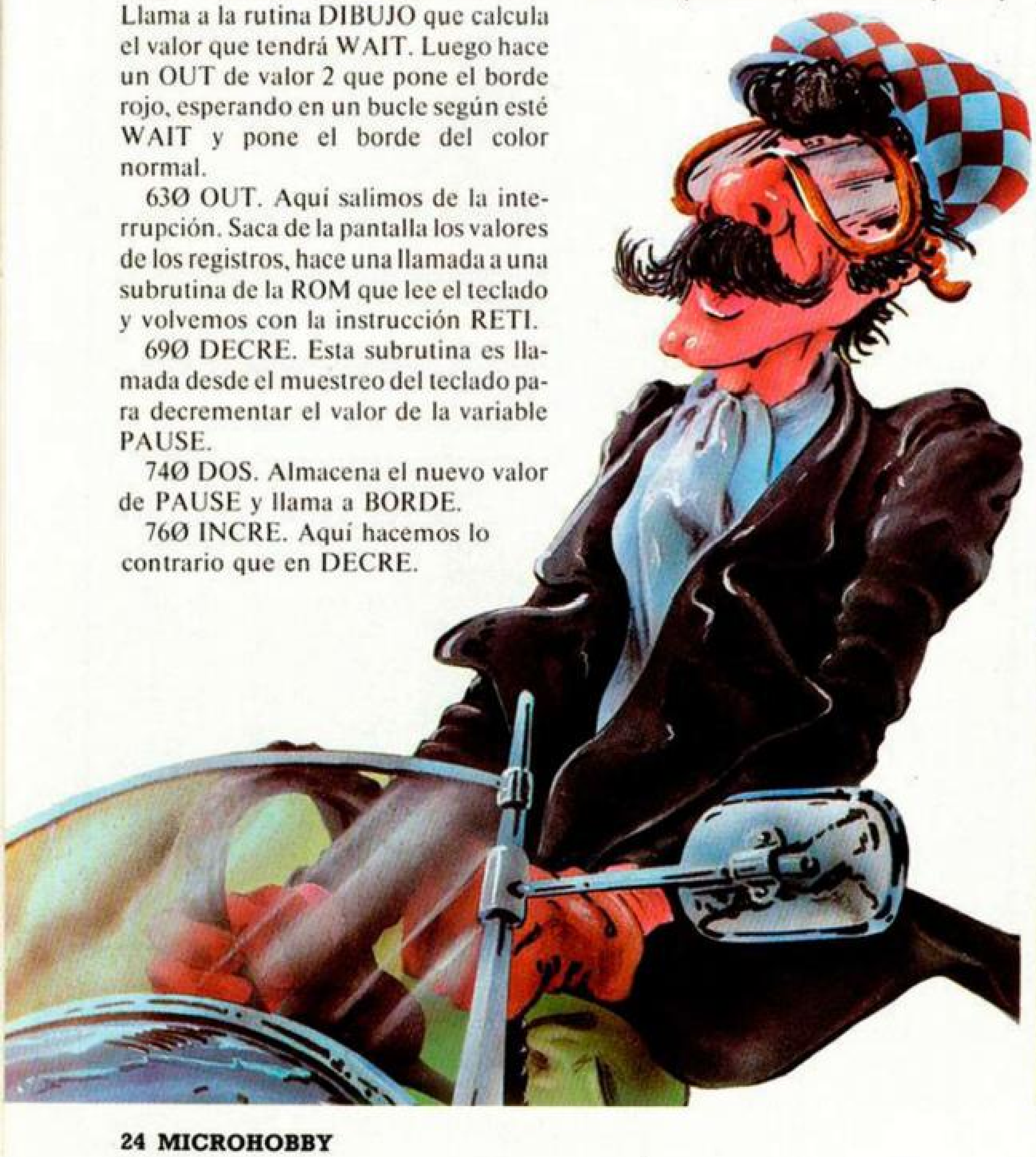
125Ø BORDEN. Esta subrutina es la que produce la pausa cuando tenemos la interrupción en el estado con pausa y sin borde. Es distinta a la que normalmente se usa (PAUSA) porque al no tener que hacer los OUT's y retardos se necesita más pausa. También muestrea el teclado por si pulsamos la combinación de teclas para salir de este estado y volver al normal.

142Ø NEW. Aquí hacemos un NEW limpiando la memoria, pero antes actualizamos dos variables del sistema y luego llamamos a una rutina de la ROM. La interrupción no se borrará pero habrá que volver a activarla.

Hasta aquí el comentario. Para los afortunados poseedores de algún programa ensamblador publicamos el listado en Assembler.

Los que lo deseen pueden utilizar el Cargador Universal de Código Máquina publicado en nuestro número 31. Para ello, utilizar el listado 2. Una vez tecleado, hay que hacer un DUMP en la dirección 65ØØØ. Por último, con la opción SAVE, salvar el Código Objeto generado indicando 65ØØØ como dirección y 294 como número de bytes.

En cualquiera de los dos casos habrá que grabar primero en la cinta el Programa 1, que es el encargado de hacer el CLEAR y cargar el bloque de bytes en su ubicación adecuada. Para ello, una vez tecleado se salvará en cinta con SAVE «Variador» LINE 1.



PROGRAMA 1

```
  10 REM Variador de velocidad
  20 REM J.M.Lazo
  30 CLEAR 64999
  40 LOAD ""CODE
  50 PRINT "      VARIADOR DE VE
LOCIDAD."
  60 PRINT AT 0,6; OVER 1;"______
_________"
  70 PRINT "RANDOMIZE USR 65000:
"
  80 PRINT "    ACTIVA LA INTERRU
PCION."''
  90 PRINT "RANDOMIZE USR 65007:
"
 100 PRINT "   DESACTIVA LA INTE
RRUPCION."''''
 110 PRINT #0;"Pulse cualquier t
ecla para NEW."
 120 PAUSE 0
 130 NEW
```

LISTADO 2

LINEA	DATOS	CONTROL
1	3EFDED47ED5EC9ED56C9	1679
2	00000000000000000000	0
3	00000001FEF5C5D5E5C3	1334
4	8EFE3E7FDBFECB4F2029	1413
5	3EBFDBFECB4720213EF7	1374
6	DBFECB47284DCB4F2857	1273
7	CB57CAD3FE3EEFDBFECB	1934
8	47CAFDFECB4FCAB1FE18	1719
9	CFCDB9FE3E02D3FE2AD1	1631
10	FE2B7CB520FB3A485CCB	1310
11	2FCB2FCB2FCBDFCBE7CB	1610
12	9FCBA7D3FE3A86FECB2F	1690
13	47CD87FE10FBE1D1C1F1	1800
14	FFED4D3A86FE3DFE0120	1363
15	013C3286FE18BE3A86FE	1159
16	3CFE6F20F33D18F002C5	1224
17	060810FEC1C93AFCFEFE	1496
18	0028453AB8FEFE00C208	1061
19	FE3EF7DBFECB4720C1CB	1738
20	4F20BD3E013288FEC308	1054
21	FE3E003288FE18AE0006	1006
22	0D373F3A86FE26006FE5	955
23	D1ED5A10FC11EC011922	1117
24	D1FEC9FC013E0032FCFE	1535
25	188C3A86FE47C5062910	941
26	FEC110F83EF7DBFECB47	1767
27	C266FECB4FC266FE3E01	1445
28	32FCFEC366FE01218B5C	1411
29	36EF2336FA2336FF2336	1065
30	FFC3B71100000000000000	650



AREAS DE TRABAJO ASOCIADAS AL MICRODRIVE E INTERFACE 1

Luis E. JUAN

Cuando se conecta al Spectrum el Interface 1, la disposición de la memoria varía para adecuarse a las nuevas necesidades de operación con el Microdrive, la red local, etc. Es fundamental conocer la distribución de estas nuevas áreas para obtener el máximo rendimiento de nuestro periférico.

El Microdrive es un dispositivo de almacenamiento de datos híbrido entre los sistemas secuenciales y aleatorios. Aunque por el aspecto del cartucho se asemeja más a una cinta de cassette que a un disco, la forma de guardar los datos es sustancialmente distinta.

La figura 1 muestra como se almacenan los datos en Microdrives en comparación con la cinta tradicional, en sucesivas «ampliaciones».

EL MAPA DE LA RAM INFERIOR

El mapa de la zona inferior de la RAM del SPECTRUM (tanto en 16 como en 48K) es el de la figura 2. Si no se tiene conectada la Interface 1 o no se ha hecho uso de ella desde la conexión del ordenador o desde un reset, las variables CHANS y PROG contienen las direcciones de memoria 23734 y 23755, respectivamente, dado que no existen mapas de microdrive y los únicos canales presentes son los K, S y P. (Ver MICROHOBBY n.° 38). Ahora bien, en cuanto se invoca cualquiera de los comandos adicionales asociados a la Interface 1 (por ejemplo, CLS#, que simplemente borra la pantalla y pone borde y papel blancos y tinta negra), ésta introduce a partir de la dirección 23734,28 nuevas variables del sistema que absorben 58 bytes adicionales de RAM, desplazando CHANS (y el origen de los mapas) hasta 23792 y PROG a 23813.

Es conocido que algunos programas incluyen código máquina en las primeras líneas, con el formato 1 REM... seguido de códigos carentes de sentido al intentar listar el programa BASIC. Dada la forma en que el SPECTRUM almacena las líneas del programa (ver capítulo 24 del manual del ordenador), la primera posición de memoria disponible para el código máquina así almacenado es la 2376Ø (23755 + 2 bytes del número de línea + 2 bytes con la longitud de la línea + 1 byte código «234» correspondiente al comando REM). Así pues, todo programa que contenga código máquina almacenado de esta manera será incompatible con la Interface 1 (y, por supuesto, con el microdrive), a no ser que el código sea reubicable y las llamadas a dicho código desde BASIC sean del tipo «... USR (5 + PEEK 23635 + 256 * PEEK 23636)», que tiene en cuenta el desplazamiento sufrido por la variable PROG (58 bytes).

Independientemente de la circunstancia anterior, toda operación que involucre un microdrive acarrea la ocupación provisional de otros 627 bytes de memoria comprendidos entre la dirección 23792 (nuevo origen de los mapas del microdrive) y el programa BASIC, que sufre un desplazamiento hacia arriba en memoria mientras dura la operación. Estos 627 bytes corresponden a las siguientes áreas de trabajo:

— 32 bytes configuran el mapa del cartucho en uso, representando cada bit a un sector (en total, 32 x 8 = 256 bits de los que sólo se vienen a usar un máximo de 2ØØ). Si el bit vale «1» dicho sector no es utilizable.

— 595 bytes configuran un canal de microdrive: 512 corresponden a buffer de datos (para un sector) y los 83 restantes almacenan parámetros relativos a identificación del sector y archivo en uso (ver Apéndice 3 del manual del microdrive).

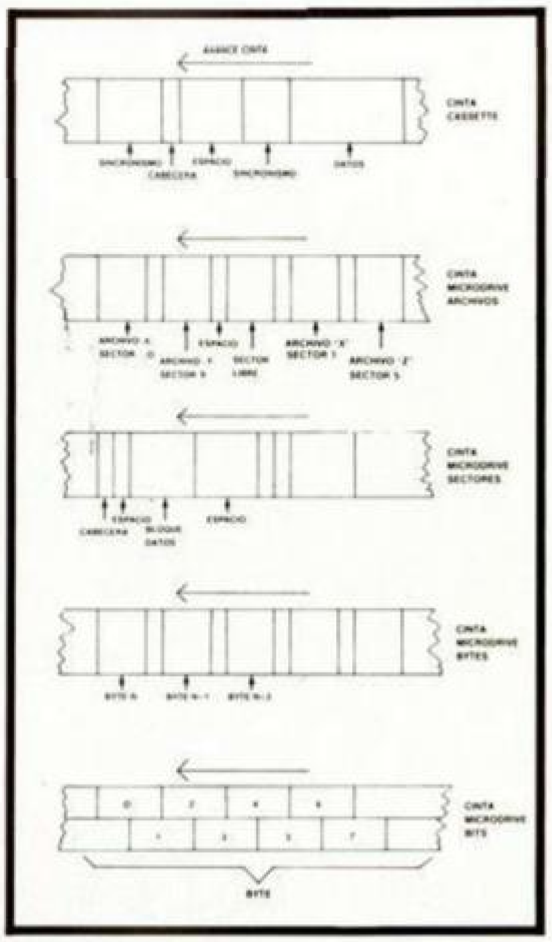


FIGURA 1

Así pues, la primera posición de memoria disponible para el BASIC durante la operación del microdrive será la 23755 + 58 + 627 = 2444Ø. Este hecho es fundamental en la conversión de programas originalmente almacenados en cassette para su paso a microdrive.

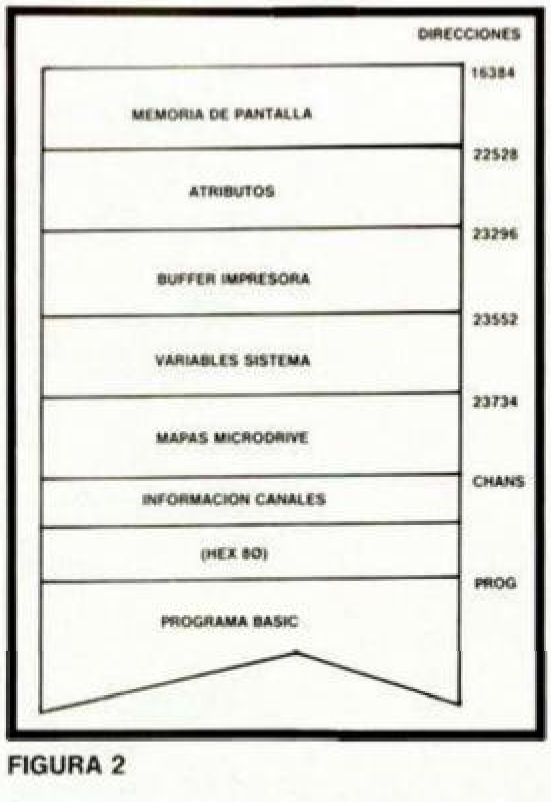
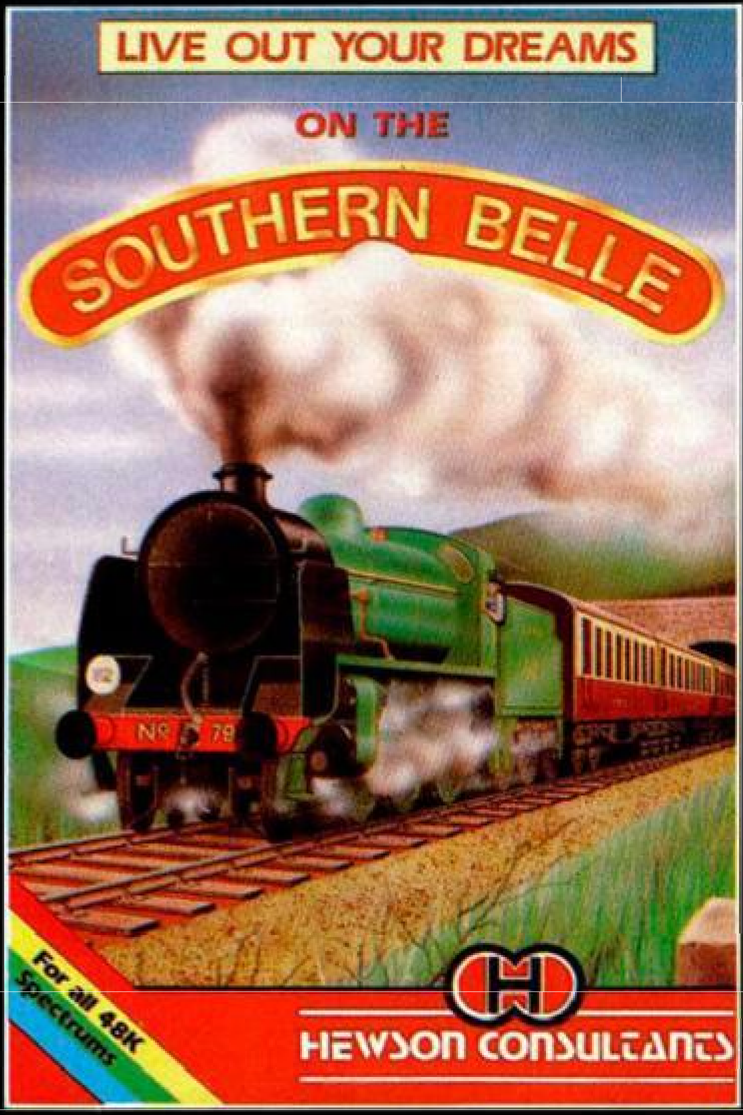


FIGURA 2



SOUTHERN BELLE

BIENVENIDOS A BORDO, OCUPEN SUS ASIENTOS Y PREPARENSE PARA UN EMOCIONANTE VIAJE EN "LA BELLA DEL SUR" LA MARAVILLOSA LOCOMOTORA DE LOS AÑOS 20.
CON ESTE PERFECTO SIMULADOR, CONTROLARAS TODOS LOS APARATOS DEL TREN, LA PRESION DEL VAPOR, EL CARBON DE LA CALDERA, LAS SEÑALES QUE ENCUENTRES EN LAS VIAS Y HASTA EL SILBATO QUE TENDRAS QUE USAR EN LOS MOMENTOS OPORTUNOS.
CONTROLAR UN TREN NO ES COSA FACIL Y PARA AQUELLOS QUE OS GUSTEN LOS SIMULADORES DE VUELO, OS VENDRA BIEN PONER LOS PIES EN TIERRA Y APRENDER A MANEJAR LA LOCOMOTORA QUE MARCO TODA UNA EPOCA.
SPECTRUM/AMSTRAD

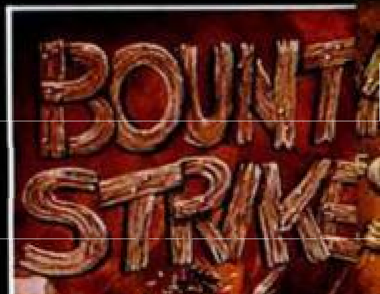


FIGHTING WARRIOR

COMO FIGHTING WARRIOR CAMPEON DEL ANTIGUO EGIPTO, DEBES USAR TODOS TUS PODERES Y TECNICAS DE COMBATE EN EL DESAFIO QUE TENDRA COMO PREMIO EL RESCATE DE LA PRINCESA ENCERRADA EN LA GRAN PIRAMIDE. PARA LLEGAR A ELLA HABRAS DE MANTENER DUROS COMBATES CUERPO A CUERPO ARMADO SOLO CON TU ESPADA, QUE DEBERAS MANEJAR DE FORMA EXPERTA CONTRA CRIATURAS MITAD HOMBRE, MITAD DEMONIO QUE SURGIRAN, A TU PASO. SENSACIONES GRAFICOS Y MOVIMIENTOS CREADOS POR LOS MISMOS, QUE HICIERON "EXPLODING FIST" UNIDOS A UNA GRAN CANTIDAD DE SORPRESAS HARAN DE FIGHTING WARRIOR TU JUEGO PREFERIDO.
SPECTRUM.

ERB
Soft

¡TI
LO
EXIT



BOUNTY BOB

SI TE GUSTAN
"PLATAFORMA" (
MINE
NO LO DUDE
ES TU
NO TE DECIMOS M
FAVORITA Y DILES
¡SEGURO Q

DISTRIBUCION EXCLUSIVA PARA ESPAÑA ERBE SOFTWA

ware

ENE
OS
TOS!

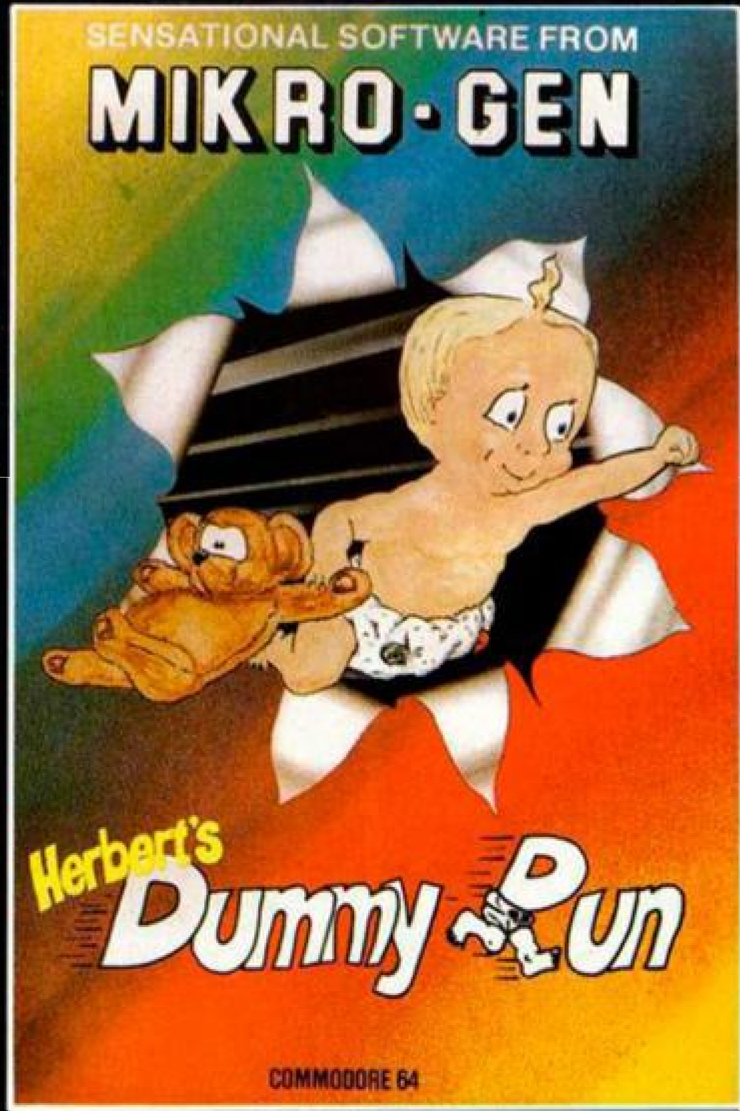


TRIKES BACK

OS JUEGOS DE
U SIMBEL, MANIC
E ETC.)
S BOUNTY BOB
JEGO.
S, VE A TU TIENDA
UE TE LO ENSEÑEN,
E TE GUSTA!

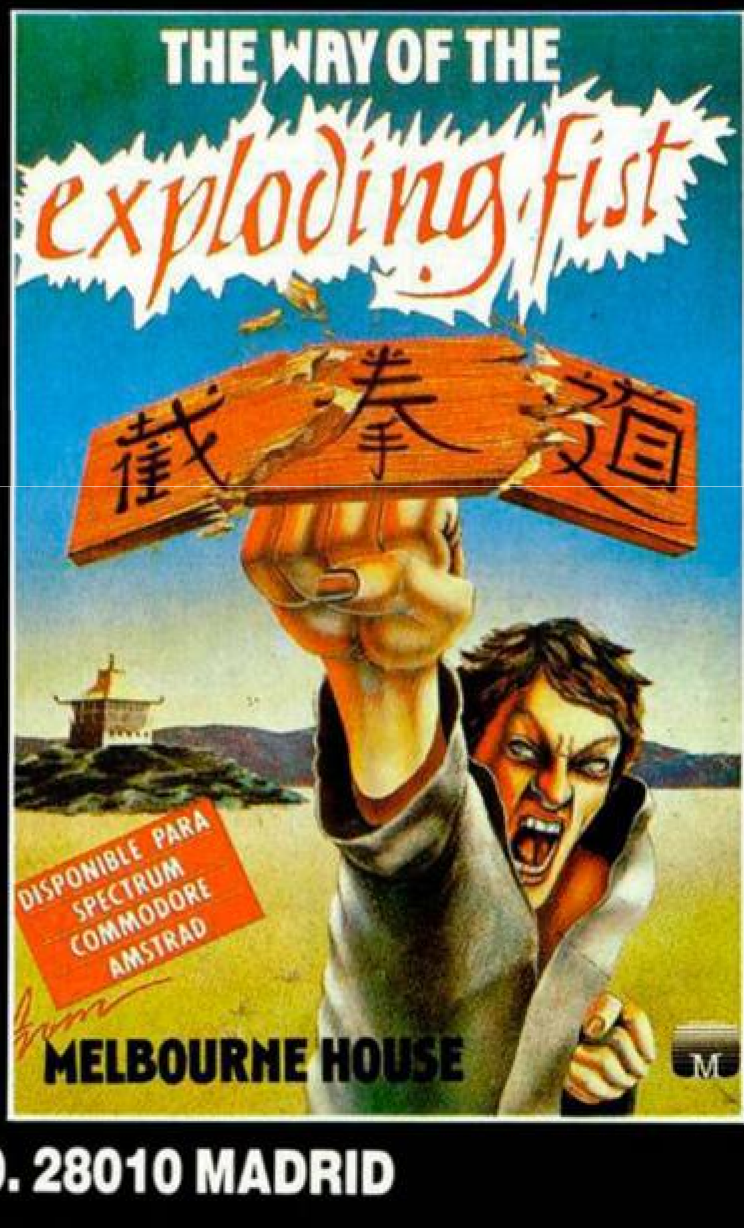
HERBERT'S DUMMY RUN

LA CRITICA HA DICHO: HERBERT'S ES UNA MEZCLA ENTRE EL "PYJAMARAMA" Y EL "EVERYONE'S A WALLY" DE LOS QUE HA ESCOGIDO LO MEJOR PERO AUMENTANDO LA CALIDAD DE GRAFICOS Y MOVIMIENTOS. RECOMENDADO PARA TODO TIPO DE USUARIOS, LOS QUE PREFIERAN LOS JUEGOS SENCILLOS Y LOS QUE QUIERAN JUEGOS UN POCO MAS COMPLICADOS EN LOS QUE ADEMAS DE SER HABILIDOSO QUIERAN ACTUAR CON LOGICA Y CREAR UNA ESTRATEGIA A SEGUIR. Y SOBRE TODO RECOMENDADO PARA LOS QUE DISFRUTARON CON PYJAMARAMA Y WALLY
SPECTRUM / COMMODORE / AMSTRAD



EXPLODING FIST

EL MEJOR JUEGO DE KARATE QUE PUEDAS ENCONTRAR. EL PROGRAMA MAS VENDIDO ACTUALMENTE EN INGLATERRA. 18 GOLPES Y MOVIMIENTOS DISTINTOS CUIDADOSAMENTE DISEÑADOS PARA SIMULAR UN EMOCIONANTE COMBATE EN EL QUE TODOS TUS REFLEJOS Y HABILIDADES SERAN PUESTAS A PRUEBA.
SPECTRUM / COMMODORE / AMSTRAD



RE, SANTA ENGRACIA, 17, TEL. 447 34 10. 28010 MADRID

EL MICROPROCESADOR Z-80 (I)

Primitivo de FRANCISCO

En el interior del Spectrum exite un componente frecuentemente mencionado, que es el elemento inteligente capaz de interpretar y ejecutar las instrucciones, siempre en código máquina, que le proporcionamos mediante el intérprete Basic o mediante código máquina directamente. Del mayor conocimiento posible del Z-80 depende también la perfecta comprensión de futuros montajes que además, requerirán un cierto volumen de Software.



Aspecto físico del Microprocesador Z-80, cerebro del Spectrum.

Se dice que el Z-80 es un microprocesador de la tercera generación y, en efecto, surgió como microprocesador de ocho bits que acumulaba en sí todos los perfeccionamientos del momento más los que se le añadieron.

El Z-80 sale al mercado en el año 1976, desarrollado por una nueva firma americana llamada ZILOG, y creada por un grupo de técnicos que habían trabajado anteriormente en INTEL CORPORATION en donde se había desarrollado dos años antes el famoso microprocesador 8Ø8Ø, que fue el primero de ocho bits que obtuvo un éxito total a escala comercial e industrial.

En estas condiciones ZILOG, con la experiencia anterior acumulada que combinaba el conocimiento del diseño y fabricación de microprocesadores con la arquitectura y software de ordenadores, y con el soporte financiero del grupo EXXON americano, creó el Z-80 cuyas siglas corresponden: Z a ZILOG y 8Ø porque es un digno hijo del 8Ø8Ø a quien imita, compatibiliza y supera.

Sobre un chips cuya superficie es un 20% mayor que la del 8Ø8Ø, el Z-80 integra cerca de 8.000 transitores (Aproximadamente 4.500 más que el 8Ø8Ø). También hay que decir que posee 158 instrucciones (696 con los diferentes modos de direccionamiento). Tras la comercialización del Z-80 y otros, el 8Ø8Ø fue quedando anticuado, permaneciendo su producción relegada a mero repuesto de los equipos ya existentes. El abandono del 8Ø8Ø también fue provocado por la propia INTEL que poco después desarrolló el 8Ø85 con notorias mejoras, respecto a su predecesor, el 8Ø8Ø.

El desarrollo de los microprocesadores de ocho bits no ha progresado posteriormente porque los fabricantes se han dedicado prioritariamente a potenciar sus nuevos microprocesadores de 16, y últimamente, 32 bits. Es por esto y por su avanzada concepción por lo que el Z-80, pese a tener casi diez años de ancianidad, sigue en cartel en el campo de los ocho bits, y hoy lo fabrican además de ZILOG: MOSTEK, SGS, NEC y SHARP.

Caracterícticas del Z-80

El microprocesador Z-80 está fabricado con tecnología MOS canal N, puerta de silicio. La tecnología MOS es la más utilizada para la construcción de microprocesadores, dado que es la única que permite una gran integración a bajo costo, lo que da lugar a un reducido consumo de corriente y consecuentemente a una necesidad mínima de disipación de calor. Su capacidad de direccionamiento es de 64 Kbytes proporcionados por sus 16 bits de dirección.

La frecuencia del reloj es de 2'5 MHz y 4 MHz para la versión Z-80-A. Esta señal de reloj precisa de una única fase (Φ), lo que se traduce a un solo terminal de entrada de reloj, el cual ha de ser exterior al microprocesador.

Su encapsulado es el conocido DIL (Dual In Line) de 40 patas.

Su alimentación es únicamente de 5 voltios frente al 8Ø8Ø que necesitaba además +12 V y −5 V. Posee 158 instrucciones en código máquina y sus modos direccionamiento son: Implícito, inemdiato, relativo, directo, por registros e indexado.

Posee dos modos diferentes de interrupción no enmascarable y enmascarable, este último con tres facetas.

El Z-80 contiene un total de 22 registros de 16 y 8 bits, algunos de los de 16 bits se pueden desdoblar en dos de ocho.

Sus bus de dirección es de 16 bits y el de datos de ocho.

Su hardware externo está reducido al máximo pudiendo ser conectado a memorias RAM estáticas o dinámicas, para estas últimas dispone en su interior de una lógica completa de refresco.

Funcionamiento del Z-80

Excepto en sus particularidades ya mencionadas, el Z-80 no se diferencia sustancialmente del resto de los microprocesadores ya existentes en cuanto a su filosofía de funcionamiento; únicamente su amplia bancada de registros merece un detenido comentario.

Para empezar hay que decir que toda la dinamicidad de un microprocesador depende de su entrada de reloj (Clock) que en el Z-80 es de un único terminal de entrada por donde se aplicará desde el exterior una señal de niveles TTL y a la frecuencia mencionada anteriormente. Este «clock» es el corazón del microprocesador cuyos latidos proporcionarán el ritmo de interpretación, ejecución y cálculo de las instrucciones que le llegan desde la memoria en donde han de estar contenidas previamente.

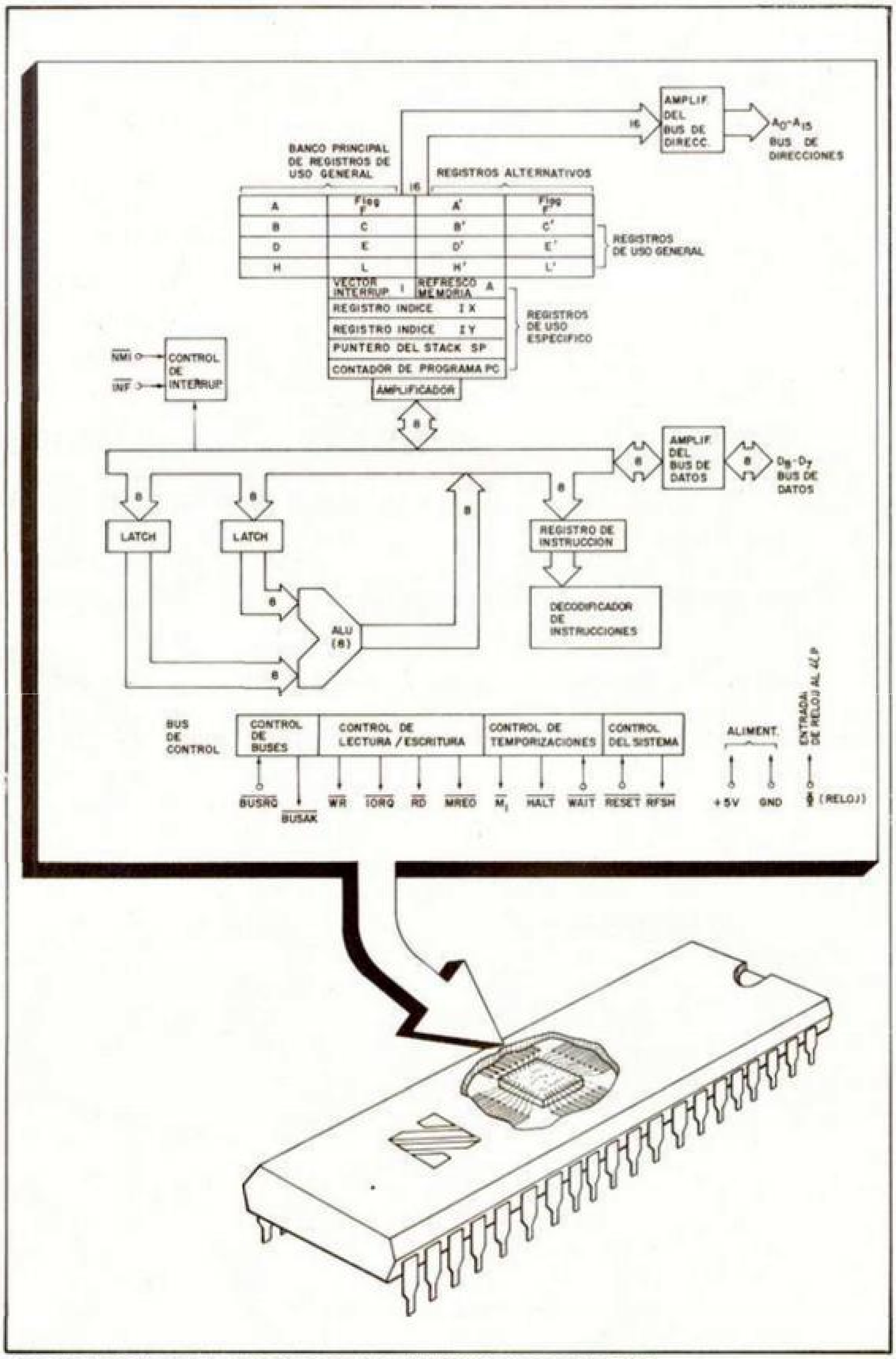
Por tanto, cuanto más alta es la frecuencia de reloj de un microprocesador, más rápido será en su ejecución,

(siempre naturalmente dentro de márgenes establecidos por el fabricante).

En el interior del Z-80 dos son los bloques encargados de interpretar y procesar las instrucciones que se le suministran ordenadamente; estos bloques son: el decodificador de instrucciones y la ALU (Siglas en inglés cuya traducción es UNIDAD ARITMETICO-LOGICA). El decodificador interpreta cada byte que le llega por el BUS de datos y activa en consecuencia los diferentes circuitos en función de una programación previa que ha introducido el fabricante en este bloque. Cada una de las posibles decodificaciones se corresponde con el primer byte del código máquina que suministra el fabricante en las tablas de instrucciones del microprocesador. Dado que el decodificador es de ocho bits, sería posible decodificar hasta 256 instrucciones; pero, puesto que en el Z-80 se ha querido disponer de hasta 696 posibilidades de interpretación, se hizo necesario ampliar a dos bytes el código de algunas instrucciones, este es el caso de aquellas cuyo primer byte es en hexadecimal: CB, DD, ED y FD. En principio parece que la codificación de instrucciones la confecciona el fabricante caprichosamente, pero si se hace un desglose de sus códigos se observa cierto orden funcional. Veamos un ejemplo: Una de las instrucciones más simples es cargar el registro "A" con un dato, su nemónico es LD A, 5D (5D es el dato). Su codificación en memoria quedaría: 3E 5D en donde 3E es el código que se corresponde con la orden: "carga el registro A con el dato 5D que viene a continuación". Un ejemplo de codificación de dos bytes sería: LD A, (IY + Ø7) que quiere decir: "carga el registro A con el dato que apunta el registro IY cuya dirección está formada por el contenido del registro IY + el byte que le sigue llamado desplazamiento", esta instrucción sería en código máquina FD 7E Ø7 en donde FD y 7E conforman los dos bytes de código y Ø7 es el desplazamiento.

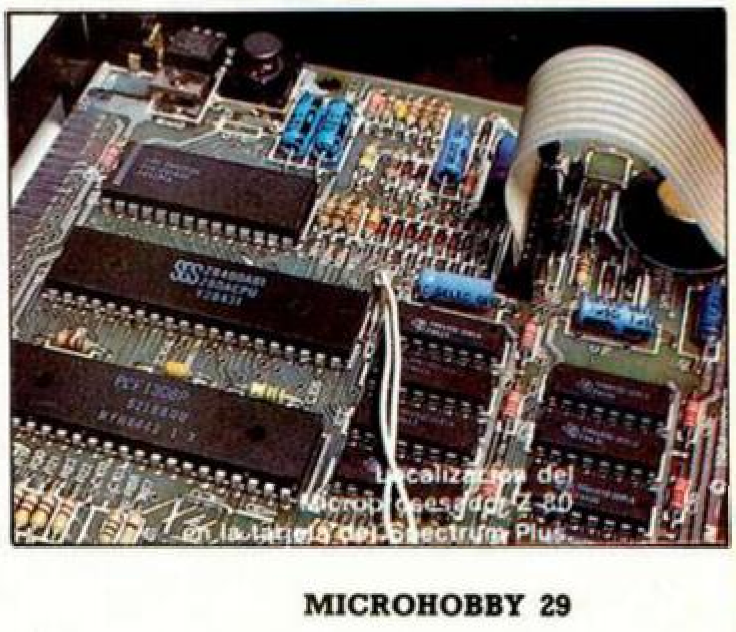
La ALU es el bloque encargado de sumar, restar o hacer operaciones lógicas entre dos bytes y entregar el resultado por el registro A o HL afectando al registro de estado o registro de Flags denominado F. Los dos bytes a procesar han de ser previamente cargados en los LATCH de sus entradas para que estén presentes mientras opera la ALU. Todos estos procesos son controlados por la lógica interna accionada por el decodificador de instrucciones bajo la acción del reloj, que como hemos dicho, encadena toda esta cascada de acontecimientos.

En el interior del Z-80 dos son los registros que desempeñan un papel importante en la ejecución de un programa, el contador de programa (PC) y el registro de estado (F). El contador de programa es de 16 Bits y tiene, por tanto, la capacidad de acceder a todos los bytes de la memoria que serán un total de 65.536 (en hexadecimal desde ØØØØ hasta FFFF). Este registro lleva



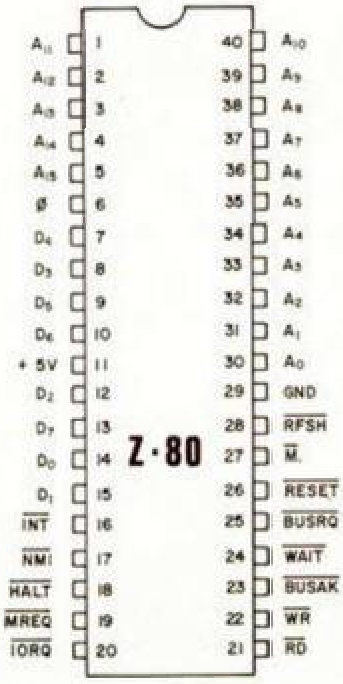
Aspecto exterior y bloques funcionales del Microprocesador Z-80.

Localización del Z-80 en el interior del Spectrum Plus.



la dirección de memoria de cada uno de los bytes de la instrucción en curso y se incrementa de uno en uno cada vez que el decodificador de instrucciones va tornando de la memoria los bytes que va interpretando. Se ve incrementado al final de cada instrucción tantos bytes como contenga cada instrucción. En el momento del reset del microprocesador todos los registros se ponen a cero, es por eso que el arranque del Spectrum, tras un reset o un encendido, siempre se inicia por la ROM que está direccionada desde ØØØØ hasta 3FFF en hexadecimal (16384 bytes).

El Status, registro de estado o de flags indica en todo momento el resultado operativo del microprocesador lo cual sirve al programa principalmente para tomar decisiones y efectuar saltos condicionados. Pongamos un ejemplo: Si se efectúa una OR exclusive del registro A consigo mismo el bit Z del status se pone a uno indicando que el resultado de esta operación ha sido un cero. En efecto, un número en OR exclusive consigo produce siempre ØØ H. Similarmente si un dato se suma con otro y se produce un acarreo, entonces el bit se pondrá a uno, circunstancia ésta que posibilita la bifurcación del programa a partir de este punto si es que lo requiere la concepción del programa.



Distribución de señales en los terminales del Microprocesador Z-80.

En todo momento el Z-80 se comunica con la memoria y periféricos mediante tres buses de señales que son: El bus de **direcciones**, el de **datos** y el de **control**. El bus de direcciones presenta hacia el exterior la dirección del byte en proceso o la dirección del periférico habilitado. Esta compuesto de 16 bits que se numeran desde AØ hasta A15. El bus de datos compuesto de ocho bits desde DØ hasta D7, por él discurren datos en hexadecimal para leer de memoria o para escribir en ella, según las directrices del programa. Este hecho obliga a que este bus sea de doble sentido lo cual conlleva algunas complicaciones Hardware como veremos, pero que a cambio simplifican líneas de cableado que usan este bus. El bus de control es un poco específico de cada microprocesador. En el caso del Z-80, este bus se puede desglosar bajo cuatro conceptos: el control de buses, el control de los procesos de escritura y lectura, el control de las señales de temporización y el control del sistema.

Existe también un bloque anexo para el control de interrupciones y circuitos amplificadores en los buses que permiten un mayor FAN OUT adaptando todas las señales a los niveles TTL.

En cuanto a los registros existen además de los dos ya mencionados, dos bancos intercambiables de uso general de 6 registros cada uno, un registo acumulador, dos registros índice de 16 bits, un puntero de stack y dos registros de ocho bits para el control de refresco y para el vector de interrupciones. (Ver figuras). De todo esto trataremos en próximos apartados hardware.



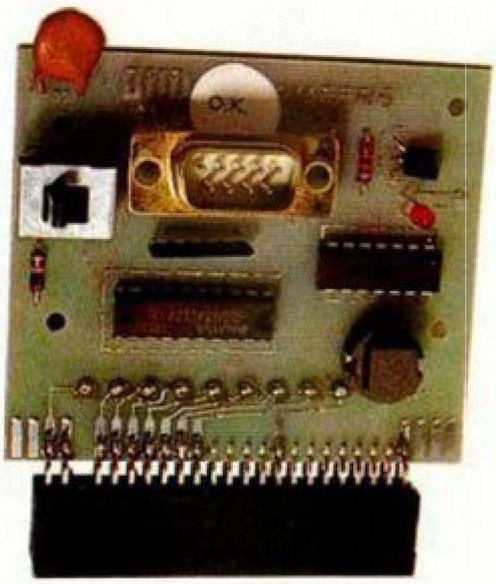
INTERFACE DE JOYSTICK «INTERSTATE 31»

Un nuevo interface de Joystick ha salido al mercado a un precio de 1.900 pesetas distribuido en España por Sinclair Store. Se trata de la «Interstate 31», de la firma «Buda Computers». Tiene protocolo Kempston y puede leerse desde Basic usando la función IN con el port 31, de ahí el nombre de «Interstate 31».

Aparte del cometido principal de conectar el joystick, tiene también incluido un botón de «reset» y un piloto indicador de encendido.

También lleva incorporado un conmutador de disparo normal





o automático. Es decir, disparo sin necesidad de mantener pulsado el botón del joystick.

Con la ayuda de este sencillo programa podemos saber el resultado de la función IN en las distintas posiciones de la palanca, lo que nos servirá para adaptar nuestros propios juegos en BASIC.

```
10 PRINT AT 0,0; "IN 31: "; IN 31; " ": GOTO 10
```

Memoria Bloqueada

Soy un poseedor de un ZX Spectrum Plus, y no sé por qué motivo, el ordenador ya no me carga programas largos, a la vez que me sale el mensaje "M RAMTOP no good".

I.A. - Guipuzcoa

□ Es posible que su ordenador tenga parte de la memoria superior bloqueada por una avería. Puede comprobarlo haciendo: PRINT PEEK 23733 que deberá dar como resultado 255. Si le diera un número inferior, la avería puede estar en cualquiera de los chips que constituyen o direccionan los 32 Ks superiores de memoria.

Protección «Turbo»

Quisiera saber si el programa «TRANS EXPRESS» de la firma ROMANTIC ROBOT es capaz de copiar los programas protegidos con sistema «Turbo».

Carlos L. SALVADOR - Madrid

□ No tenemos noticias de ningún copiador que sea capaz de copiar los programas protegidos con sistema «Turbo», ni siquiera el TRANS EXPRESS.

«MODE» y «FLAGS2»

¿Por qué el manual se refiere a la dirección 23617 como la que especifica el modo del cursor, y luego resulta que es la 23658?

¿Podríais publicar trozos de la ROM del Spectrum desensamblada?, no consigo nada parecido por ninguna parte.

Rafael PASSARO - Canarias

□ La dirección 23617 almacena la variable MODE que especifica el modo del cursor, según el siguiente orden: Ø="K", 1="E", 2="G". La variable FLAGS2, en la dirección 23658, contiene diversos indicadores o "Flags" entre ellos, el que determina las mayúsculas y las minúsculas, concretamente el bit 3, por tanto, Ø="L" y 8="C". Ambas variables afectan al cursor, pero la que realmente determina el "Modo" es la 23617.

Respecto al desensamble de la ROM, le recomendamos el lidro de los doctores Ian Logan y Frank O'Hara: "THE COMPLETE SPECTRUM ROM DISASSEMBLY", ed. Melbourne House, 1983.

Grabación de variables

He perdido el manual y no encuentro solución a este problema: Si grabo una variable alfanumérica a$ como SAVE "x" DATA a$() y luego hago un LOAD "x" DATA a$( ). ¿Qué debo hacer para trabajar con esa variable, ya que el ordenador ejecuta el LOAD a la perfección? da el conocido mensaje "OK, Ø:1", y a la hora de hacer, por ejemplo, un PRINT a$ me sale "2 Variable not found".

¿Es posible comerle memoria a la zona de gráficos para ampliar la zona de Basic? ¿A qué dirección hay que hacer el CLEAR?

Paco ESTEVEZ - Bilbao

□ Lo que usted comenta es uno de los fallos del Sistema Operativo del Spectrum; en realidad, el ordenador no debería salvar la variable a$ como si fuera una matriz, sin embargo lo hace, incluso la carga; el problema es que luego es incapaz de trabajar con ella.

Con este mismo problema nos encontramos nosotros cuando desarrollamos el cargador de código máquina de MICROHOBBY, en esa ocasión, lo solucionamos diseñando una rutina en C/M que busca la matriz cargada, y le altera el código indentificador para convertirla de nuevo, en una variable normal de cadena. Si estudia este programa, tal vez nuestro método le sea de utilidad.

Otra posibilidad es dimensionar la matriz antes de grabarla. En este caso, trabajaría con longitud fija. Elija el sistema que más le convenga.

Respecto a la forma de «cargarse» los UDG, debe hacer CLEAR 65535 si su modelo es de 48K, y CLEAR 32767 si es de 16.

Varias preguntas

¿Podríais decirme brevemente en qué consiste el software y en quié consiste el hardware?

¿Qué es una matriz?

¿Cómo puedo averiguar en un programa cualquierá su inicio y su longitud?

Y por último: He observado que en mi Spectrum se está desconchando su parte derecha y tengo miedo de

que afecte al lugar que ocupan las teclas. ¿Qué puedo hacer para remediarlo? ¿Se debe al calentamiento?, si no tuviera solución, ¿valdría la pena cambiar el teclado?

Joaquín RIPOLL - Barcelona

□ "Hardware" es una palabra inglesa que designa la parte tangible de un sistema informático, se puede traducir al castellano como "soporte físico". "Software" designa la parte intangible o inmaterial, es decir, los programas y datos almacenados como información; su traducción al castellano es "soporte lógico". Si lo deseas, puede asimilarlo con los conceptos de "cuerpo" y "mente".

Una matriz es una tabla de datos, ordenada en una o más dimensiones, en la que cada elemento se puede referenciar con el nombre de la matriz, seguido de uno o más subíndices (tantos como dimensiones). Imagine un tablero de ajedrez, en cada cuadro almacena un número, ahora, tiene una matriz numérica de dos dimensiones; cada dimensión tiene 8 valores posibles (digamos, vertical y horizontal), si sitúa el origen en el ángulo superior izquierdo, puede indicar la posición de cualquier elemento con dos números, la fila y la columna; estos serían los subíndices. La pantalla del Spectrum, es en realidad, una matriz bidimensional de 32 por 24 posiciones de carácter.

La dirección de inicio de un programa, puede averiguarla con: PRINT PEEK 23635+256•PEEK 23636 y la longitud con: PRINT (PEEK 23641+256•PEEK 23642) – (PEEK 23635+256•PEEK 23636).

El deterioro de su teclado, es posible que se deba tanto al calentamiento, como al uso que haga de él; en cualquier caso, siempre es interesante sustituirlo por otro de más calidad, entre otras cosas, por que es más cómodo de utilizar.

Cómo desmontar el teclado

Me dirijo a ustedes para que me expliquen la manera de desmontar el teclado del Spectrum, ya que al mio no le funcionan la "B" ni la "N".

José L. ABENIA

□ Si su Spectrum es un Plus, podrá desmontar el teclado retirando los tornillos que hay por debajo de la tapa superior, una vez abierto el ordenador.

Si por el contrario, se trata de la versión con teclado de goma, puede desarmarlo despegando la superficie metálica superior, con sumo cuidado para no doblarla.

En ambos casos, deberá realizar la operación con el teclado desconectado del ordenador, lo que se logra desenchufando las dos cintas de 5 y 8 conductores respectivamente.

Dibujo artístico

Me gustaría realizar pantallas de presentación como las de los programas comerciales. He visto que existen en el mercado varios programas para ello: Artist, Supergráficos, etc. ¿Cuál de ellos me aconsejan?

Jesús CID - Tarragona

□ El programa "Supergráficos", es la versión española del conocido "Melbourne Draw", el "Artist" es de producción nacional. Ambos son excelentes programas de dibujo, si bien, nuestras preferencias están más del lado de "Supergráficos".

Interfaces programables

¿Es programable el INTERFACE 2?

En caso negativo, ¿cuál es el precio aproximado de un interface programable?

¿Qué utilidad pueden tener los cartuchos de ROM?

José M. GOMEZ - Murcia

□ El INTEFACE 2 no es programable, los joystick actúan sobre los mismos ports que la fila superior del teclado.

El precio de un interface de joystick programable ronda las 10,000 pesetas.

El INTERFACE 2 fue pensado para convertir al Spectrum en una consola de video-juegos, en esta línea, los cartuchos de ROM deberían contener los juegos propiamente dichos. El caso es que los cartuchos salen demasiado caros, y los fabricantes no se han animado a producir software en este soporte, por lo que su utilidad ha quedado muy reducida.

DE OCASION

● DESEARIA contactar con grupos de chicos y chicas de Madrid para intercambiar ideas, trucos, información, etc. Mi dirección es la siguiente: Antonio Pérez. Lope de Haro, 22. Madrid 28039 o bien llamar al Tel.(91)2702623.

● VENDO/CAMBIO la cinta de demostración «Horizontes» (original) y la del Spectrum Plus (copia) por 3.000 ptas. o bien lo cambio por Interface o Joystick. Interesados contactar con Víctor Montañana. Democracia, 2, 1.°. Pobla de Vallbona (Valencia). Tel. (96)2760268.

● VENDO ZX Spectrum 48 K, con revistas de información por el importe de 30.000 ptas. Llamar el Tel. (93)3517789 (a partir de las 21 h), o bien escribir a la siguiente dirección: Esteban Salas Pereira. Desfar, 25, 3.°. Barcelona 08016.

● CLUB de usuarios del Spectrum acepta nuevos socios. No dudes, llámanos para informarte. Tenemos una revista mensual. Si te interesa puedes llamar al Tel. 2696136 de Madrid, preguntar por Jaime.

● VENDO impresora térmica Sinclair por 10.000 ptas. Calculadora programable Casio FX-702P por 15.000 ptas. Llamar al Tel. (942)373898 (noches). Preguntar por Vicente Díaz.

● VENDO ZX Spectrum Plus nuevo, completo con instrucciones en castellano e inglés, más interface programable, joystick Quick Shot I. Todo está en garantía, todo por 35.000 ptas. Interesados llamar al Tel. (93)7920853 (10 a 13 y 16 a 20 h). Preguntar por Manuel Real.

● DESEAMOS formar un club de Hardware a ser posible de Madrid. Se valoran conocimientos en microelectrónica, código máquina, inglés o simplemente saber bricologe. También buscamos documentación. Escribir a Teófilo Fernández. Biarritz, 3. Madrid. Tel. (91)2567462.

● VENDO joystick Spectravideo II e interface Kempston, nuevos por el precio de 5.000 ptas. Jesús M.ª Hernández. Ofra Cobasa bl. 9-11, 1.° derecha. Santa Cruz de Tenerife.

● VENDO video-juego Atari completo, con 4 cartuchos por 10.000 ptas. (negociables). También lo cambio por un teclado profesional, ZX 81, un cassette para ordenadores o por un Currah microspeesh. También vendo Interface programable con joystick, por 55.000 ptas. o bien lo cambio por algo de lo anterior. Me interesa además formar un club a nivel nacional. Interesados escribir a J. Ramón Jofre Sainz de la Maza. Avda. Extremadura, 4, 2.° A. San Sebastián de los Reyes (Madrid). Tel. 6542029.

● SOMOS el ZX Castellón Club y admitimos socios, para todo tipo de intercambio: trucos, información, ideas, etc. Si te interesa contacta con nuestro club, escribiendo a la siguiente dirección: Club Castellón (José M.). San Francisco, 52, 4.°. Alcora (Castellón).

● VENDO Spectrum 16 K, con manual en castellano, revista en cassette por el precio de: 21.000 ptas. Dirigirse a Rafael Fuertes. Mayor, 13. Andosilla (Navarra). Tel. (948)674056.

● VENDO Spectrum de 48 K, con TV b/n de 22", interface programable de Indescomp, joystick, cassette especial para ordenador (cuenta-vueltas, auto stop, luz, pausa, etc.). Todo por 55.000 ptas. Llamar al Tel. 6529851 de Madrid.

● VENDO Spectrum 48 K, en muy buen estado, con cables, fuente de alimentación, interruptor ON/OFF, reset, manual en inglés y español, cintas cassette, libro de Código Máquina, revistas. Por el precio de 30.000 ptas. Teclado profesional para el Spectrum por 7.000 ptas. y todo en conjunto por 35.000 ptas. Dirigirse a Miguel Moreno. Cataluña, 48, 3.°. El Prat (Barcelona).

● DESEAMOS contactar con usuarios de Sevilla y Provincia (preferentemente), para intercambio de trucos, información, utilidades, etc. Interesados llamar al Tel. (954)780054 (tardes). Manolo.

● VENDO Spectrum 48 K, completo, con garantía y cassette, «Computone». Precio: 40.000 ptas. También vendo unidad de disco «discovery 1» con garantía de 6 meses en 55.000 ptas. Interesados pueden llamar al Tel. (985)292187.

● HAGO cabeceras y protecciones. Si te interesa puedes escribir a la siguiente dirección: Miguel Alvarez Muñoz. Paseo de Ronda, 77, 3.° B. 18004 Granada. Tel. (958)255705.

● CAMBIO ZX Spectrum 48K por Amstrad CPC 464. Abonaría la diferencia de hasta 5.000 ptas. También podría prestar bibliografía. Dirigirse a Amador Merchán Ribera. Cáceres, 8, 3.° A. 28045 Madrid. Tel. (91)4674814.

● VENDO Spectrum 16K, comprado hace un año, con manuales, cables y una cinta de 60'. Lo vendo por 20.000 ptas. (negociables). Llamar al Tel. 4781562 de Madrid. Preguntar por Alberto, (llamar tardes).

● MANDANOS tu programa preferido y te lo pasaremos a Microdrive. Interesados en más información pueden llamar al Tel. (93)8860174 de Barcelona (de 13 a 14 horas). Preguntar por Mario.

● VENDO ZX Spectrum Plus, I. Kempston Joystick por sólo 34.000 ptas. Incluye garantía. Adquirido en abril del 85. Interesados en comprarlo pueden llamar al Tel. (943)601404. Preguntar por Ignacio.

● HA NACIDO un nuevo club para las zonas de La Rioja y Vizcaya. Interesados llamar al Tel. (94)6811013. Durango (Vizcaya).

● VENDO dos plantillas para Spectrum 16 o 48K, el juego «El Constructor», sólo probado una vez y los manuales del Spectrum en español, la cinta horizontes, la del Spectrum Plus de demostración, todo por el precio de 36.000 ptas. Acepto otras ofertas. Interesados escribir a Andrés Felipe Hernández López. Pza. de la Constitución, 14, 3.° Molina de Segura (Murcia). Tel. (968)611839 (de 1,45 a 3 y de 6,15 a 9,30 h.).

● VENDO Olivetti M-20 ST a buen precio, sin usar, adjunto extras (quizás impresora PR 2400), y manuales. Razón al Tel. (93)2454974. Preguntar por Jordi.

● CAMBIO teclado profesional DK'Tronics, completamente nuevo por ZX Printer o similar o bien lo vendo por 8.000 ptas. Fco. Javier Gómez Sanz. Colonia Militar de Canillejas, 13. 28037 Madrid. Tel. 7418429.

● VENDO Spectrum 48K, con todos los accesorios más interface programable para Joystick, más Joystick «Quickshot III» con teclado numérico incorporado. Todo por 32.000 ptas. Interesados llamar al Tel. (91)7293199. Preguntar por Pedro. La dirección es la siguiente: Pedro González Fernández. S. Modesto, 46, 11.° B. 28034 Madrid.

● CAMBIO por Spectrum 48K, o vendo amplificador Vieta 2001, 20W, con garantía. Se aceptan ofertas. Escribir o llamar a Jordi Espelta. Rasos de Peguera, 15, 1.° 08033 Barcelona. Tel. (93)3502499.

● DESEARIA contactar con usuarios del Spectrum para intercambio de ideas, trucos, información, etc. También me gustaría conocer a gente que estuviera interesada en el Logo en castellano del Spectrum. Interesados llamar al Tel. (94)4338020. José Ramón Reza. Santutxu, 22, 3.° A. Bilbao (Vizcaya).

● ME GUSTARIA formar un club de usuarios en Madrid, para tratar el tema de los ordenadores, intercambio de ideas, trucos, información. Interesados escribir a Eduardo de Prada Hinojo. Algodre, 6, 2.° Ecs. Izqda. 28019 Madrid, Tel. 4615311.

● VENDO vídeo-Computer Sistem Atari, 2 Joystick, 2 padle, 6 cartuchos y un transformador, todo impecable y valorado en unas 40.000 ptas., yo lo vendo por la mitad: 20.000 ptas. Interesados escribir a Isidro Aguiar García. General Km. 29, 25. Galdar (Las Palmas de Gran Canaria), o bien llamar al Tel. (928)880688.

● URGE VENDER Spectrum 48K, con cassette, cables, caja de embalaje, etc. El precio es de 48.000 ptas. No se vende por separado. Llamar al Tel.(986)781290. Preguntar por Gerardo González.

● CAMBIO los siguientes accesorios: Sintetizador de voz Currach Mespech, ZX Interface 1 o Microdrive, por las ofertas que aceptéis. Escribir a José M.ª Caballero Ruiz. González Plazas, 49. Zarza-Capilla (Badajoz), o bien llamar al Tel. (924)619052 (tardes de 7 a 8 h.).

● BUSCO el juego «The Dam Busters», a ser posible con el libro de instrucciones. Pagaría gastos de envío y fotocopias. Interesados escribir a Ernesto Sánchez. Gran Canaria, 16, 4.° 46011 Valencia. Tel. (96)3714886.

● VENDO/CAMBIO curso completo «Hable Inglés» de Sarpe por 17.000 ptas., o lo cambio por Impresora GP-50S u otra compatible con el ZX Spectrum. El curso está sin estrenar. Interesados llamar al Tel. (974)243603. Preguntar por Ricardo Fernández.

sinclair store

SOMOS PROFESIONALES

HARDWARE

	Ptas.
Spectrum 48 K + Interface + Joystick + 10 Cintas + Curso de Basic	26.700
Spectrum Plus + 10 Cintas + Curso de Basic + Joystick + Interface + Libro de Basic	36.000
QL Castellano + Monitor (Alta resolución) + Chess + Joystick + Adaptador + Libro Tasprint + QDOCTOR (Recupera ficheros)	99.750
Commodore 64 + Curso de Basic + Cassette	54.400
Amstrad 464 + Monitor + Joystick + 12 Cintas + Libro de Basic + Curso	66.900
Amstrad CPC 6128 + Monitor + Joystick + Cable de Audio + Disco S.O. CP/M Plus y Utilidades + Disco S.O. CP/M 2.2 y Logo + Disco con 6 Programas + Disco con 5 Programas de Juegos + Curso de Basic	109.500
Impresora Seikosha GP 50 S	17.990
Impresora STar Géminis 10	53.500
Impresoras Commodore	44.500
Monitores desde	19.900
OPUS DISCOVERY 1	47.500
Interruptor Switch	750
Cartuchos de Microdrive	490
Lápiz de Luz DK'Tronicks + Interface + Software	4.950

¡Recoge tu entrada para el SIMO!

JOYSTICK

Gran Capitán + Interface	3.000
Investik + Interface	3.285
Chalenger + Interface (Compatible MSX)	3.625
Quick Shot II + Interface	3.200
Interface Programable con Pilotos	3.200

TECLADOS

Teclado Saga I (el más profesional para tu Spectrum)	10.500
Teclado Multifunción Indescomp	12.600

Super ofertas soft núms. 1
Te regalamos camisetas, posters y tarjeta club Sinclair

SOFTWARE

	Ptas.		Ptas.
Hyper Sport	1.710	Basketball	1.900
Super Test	1.750	Spy Hunter	1.700
Exploding Fist	1.750	Gremlins	1.900
Dambusters	1.850	Rocky	1.500
Frankie	1.990	Beach-Head II	2.450
Polle Position	1.600	Shadow Fire	1.700
Dragonstor	1.600	Abu Simbel	1.750
Buck Rogers	1.585	Bounty Bob	1.750
West Bank	1.650	Tapper	1.700
Bruce Lee	1.700	Avalon	1.000
Everonés a Wally	1.000	Herberts	1.750
Chess the Turk	500	Fighting Warrior	1.750
Southern Belle	1.750		

Software de **Amstrad,** Commodore y Spectravídeo, 20 por 100 de descuento sobre P.V.P.
Cursos de Basic gratis al comprar tu ordenador.
Garantías oficiales en todos nuestros productos.

Aviso a los socios del Club Sinclair Store: 20 por 100 de descuento sobre P.V.P. en todos los artículos hasta el 30 de noviembre.

Bravo Murillo, 2. Tel. 446 62 31. 28003 Madrid
Diego de León, 25. Tel. 261 88 01. 28006 Madrid

PROXIMA APERTURA:
Avda. Felipe II, 12. 28009 Madrid
Tel. 431 32 33



PERIFERICOS SPECTRUM A POSTERIORI

El **Spectrum** se rodea siempre de lo mejor.
Si, a priori, te decidiste por el más popular de los ordenadores, decídete, a posteriori, por los más importantes periféricos.

Interface 1
Permite la conexión de hasta ocho Microdrives, que pueden guardar hasta 680KB de datos y programas.
Tiene un canal RS232 para conectar impresoras, modems, ...
A través de su toma de red de área local puede comunicarse con otros 63 **Spectrum.**

Microdrive ZX
Para almacenar más de 85KB en cada cartucho magnético, con un tiempo de acceso de 3,5 segundos.

Interface 2
Para usar programas contenidos en cartuchos ROM. Contiene un controlador para dos Investicks.

Investick
Joystick de gran robustez, con mando ergonómico, ventosas para fijarlo y cuatro puntos de disparo, para disfrutar más con los juegos.

Interface programable con sonido
Permite usar los Investicks en **todos** los programas, con los efectos sonoros al volumen deseado.

Lápiz óptico
Para crear directamente en la pantalla todos los dibujos imaginables. Se puede dibujar, colorear, reducir/ampliar, mezclar con textos u otros dibujos, grabar y cargar pantallas. ...

SPECTRUM A PRIORI

DISTRIBUIDOR EXCLUSIVO **investronica**

Tomás Bretón, 62 Tel. (91) 467 82 10. Telex 23399 IYCO E. 28045 Madrid
Camp, 80. Tels. (93) 211 26 58 - 211 27 54. 08022 Barcelona

etc